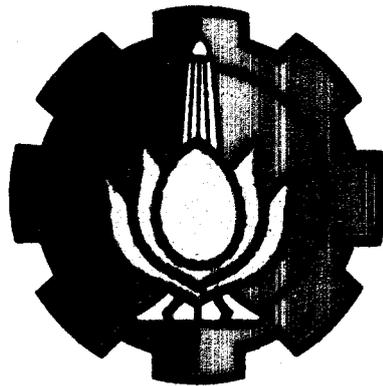


6241/173/4/94 ✓

**SISTEM KUNCI ELEKTRONIK BERBASIS  
DATA DENGAN MENGGUNAKAN  
MIKROKONTROLER 8031**

|                     |             |
|---------------------|-------------|
| PERPUSTAKAAN<br>ITS |             |
| Tgl. Terima         | 18 MAY 1993 |
| Terima dari         | TA          |
| No. Agenda          | 107 (13)    |



RSE  
G21.3916  
Pia  
S-1  
1992

**OLEH :**

**EKO PRAMUNANTO**

**NRP : 2852200378**

**JURUSAN TEKNIK ELEKTRO  
FAKULTAS TEKNOLOGI INDUSTRI  
INSTITUT TEKNOLOGI SEPULUH NOPEMBER  
S U R A B A Y A  
1992**

**SISTEM KUNCI ELEKTRONIK  
BERBASIS DATA  
DENGAN MENGGUNAKAN  
MIKROKONTROLER 8031**

**TUGAS AKHIR**

Diajukan Guna Memenuhi Sebagian Persyaratan  
Untuk Memperoleh Gelar Sarjana Teknik Elektro  
Pada  
Bidang Studi Teknik Komputer  
Jurusan Teknik Elektro  
Fakultas Teknologi Industri  
Institut Teknologi Sepuluh Nopember  
Surabaya

Mengetahui / Menyetujui  
Dosen Pembimbing



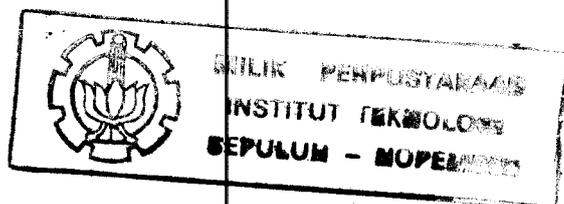
**Dr. Ir. HANDAYANI Tj. MSc**  
**NIP 130532048**

**SURABAYA  
NOPEMBER 1992**

## ABSTRAK

Kunci Elektronik Berbasis Data adalah suatu sistem kunci ruangan yang menggunakan peralatan elektronik dan komputer basis data. Sistem ini akan dapat menyelenggarakan pengamanan ruangan yang optimal.

Sistem ini bekerja didasarkan atas pengenalan peralatan elektronik terhadap kunci kartu yang mempunyai data digital yang khas. Dengan adanya data ini sejumlah kunci kartu dapat dibuat dengan data digital yang berbeda untuk masing-masing kartu, sehingga personil-personil yang berkepentingan untuk memasuki ruangan tersebut dapat membawa kunci sendiri-sendiri dan tidak terikat satu sama lain. Data dari masing-masing kartu harus disertai dengan data password yang dimasukkan lewat papan tombol untuk kemudian dilaporkan ke komputer basis data untuk disahkan dan dicatat. Dengan demikian dapat diketahui siapa saja yang telah memasuki ruangan.



## KATA PENGANTAR

Alhamdulillahirobil Alamiin. Segala puji hanya milik Allah semata. Hanya dengan rahmat serta perkenan-Nya sajalah buku tugas akhir ini dapat terselesaikan.

Tugas akhir ini merupakan karya terakhir kami dalam kegiatan perkuliahan di Jurusan Teknik Elektro FTI-ITS, yang kami harap dapat memberikan sumbangan ilmu pengetahuan yang akan dapat dimanfaatkan di kemudian hari. Tugas akhir ini juga merupakan persyaratan dalam meraih gelar sarjana di institusi ini.

Dengan segala kerendahan hati kami menyadari bahwa karya kami ini adalah karya manusia yang mempunyai sifat tidak sempurna. Oleh karenanya segala kritik dan saran membangun sangatlah kami harapkan demi kebaikan karya kami ini dan karya-karya kami selanjutnya.

Terimakasih kami ucapkan kepada semua pihak yang telah membantu demi terselesaikannya tugas akhir ini.

Surabaya, Juli 1992

Penyusun

*Dengan segala kerendahan hati  
buku ini kupersembahkan kepada  
yang tercinta :*

*Bapak Ibu*

*Istri dan Putriku tersayang*

*Adik-adikku*

## Daftar isi

|  |     |
|--|-----|
| Judul .....                                | i   |
| Pengesahan .....                           | ii  |
| Abstrak .....                              | iii |
| Kata Pengantar .....                       | iv  |
| Daftar isi .....                           | v   |
| BAB  |     |
| I PENDAHULUAN .....                        | 1   |
| 1.1 Latar belakang .....                   | 1   |
| 1.2 Permasalahan .....                     | 1   |
| 1.3 Maksud dan tujuan .....                | 2   |
| 1.4 Sistematika pembahasan .....           | 2   |
| II TEORI PENUNJANG .....                   | 4   |
| 2.1 Mikrokomputer 8031 .....               | 4   |
| 2.1.1 Pendahuluan .....                    | 4   |
| 2.1.2 Fungsi-fungsi dalam 8031 .....       | 8   |
| 2.1.3 Sistem interrupt .....               | 16  |
| 2.1.4 Timer/counter .....                  | 20  |
| 2.1.5 Port serial .....                    | 26  |
| 2.1.5.1 Baud rate .....                    | 28  |
| 2.1.5.2 Register kontrol port serial ..... | 29  |

|       |   |    |
|-------|---|----|
| 2.2   | Komunikasi serial asinkron .....                            | 31 |
| 2.2.1 | Standard interface RS-232C .....                            | 31 |
| 2.2.2 | Adapter komunikasi serial asinkron ...                      | 34 |
| III   | RANCANGAN PERANGKAT KERAS .....                             | 37 |
| 3.1   | Konfigurasi.....  | 37 |
| 3.2   | Peralatan pendukung .....                                   | 39 |
| 3.2.1 | Pembaca kartu .....   | 39 |
| 3.2.2 | Pembaca tombol password .....                               | 42 |
| 3.2.3 | Pengendali sistem mekanik .....                             | 43 |
| 3.2.4 | Pemberi tanda penerimaan penekanan<br>tombol password ..... | 43 |
| 3.3   | Catu daya .....   | 43 |
| IV    | RANCANGAN PERANGKAT LUNAK .....                             | 47 |
| 4.1   | Pendahuluan .....   | 47 |
| 4.2   | Program inisialisasi.....                                   | 49 |
| 4.3   | Rutin pelayanan interrupt .....                             | 51 |
| 4.4   | Penanganan peralatan pendukung .....                        | 52 |
| 4.5   | Pengolahan data selanjutnya .....                           | 55 |
| 4.6   | Beberapa rutin tambahan .....                               | 56 |
| V     | SISTEM BASIS DATA .....                                     | 59 |
| 5.1   | Bentuk data .....   | 59 |
| 5.2   | Penanganan data.....  | 60 |

|  |    |
|--|----|
| 5.3 Kerahasiaan data .....             | 64 |
| 5.4 Penambahan data .....              | 68 |
| 5.5 Mengubah dan menghapus data .....  | 71 |
| 5.6 Pencetakan bentuk kartu .....      | 78 |
| 5.7 Penanganan komunikasi serial ..... | 80 |
| 5.8 Program utama .....                | 83 |
| VI KESIMPULAN DAN SARAN .....          | 88 |
| VII Kepustakaan .....                  | 90 |

#### LAMPIRAN

- A Flowchart program untuk perangkat keras
- B Petunjuk pemakaian program penanganan basis data

#### TABEL

|  |    |
|--|----|
| 2.1 Alamat awal dari program pelayanan interrupt ..... | 18 |
| 2.2 Nama dan arti bit-bit pada reg. interrupt enable . | 19 |
| 2.3 Nama dan arti bit-bit pd Reg intrrupt priority ... | 21 |
| 2.4 Nama dan arti bit-bit pd Reg TCON .....            | 23 |
| 2.5 Pemilihan mode operasi dari port serial .....      | 30 |
| 2.6 Pelayanan serial port Interrupt 14H .....          | 32 |
| 2.7 Arti isi bit-bit pada reg AL .....                 | 34 |

## GAMBAR

|  |    |
|--|----|
| 2.1 Diagram blok fungsi 8031 .....   | 10 |
| 2.2 Register interrupt enable .....  | 18 |
| 2.3 Register prioritas interrupt .....                                       | 18 |
| 2.4 Reg timer/counter control/status (tcon) .....                            | 22 |
| 2.5 Register TMOD (timer/counter mode) .....                                 | 26 |
| 2.6 Register SCON .....  | 29 |
| 2.7 Kedudukan kode parameter dalam bit-bit reg AL ....                       | 33 |
| 2.8 Isi Reg AH oleh pelayanan 3 .....  | 36 |
| 2.9 Penyambungan komunikasi serial .....                                     | 36 |
| 3.1 Program memory read cycle timing .....                                   | 38 |
| 3.2 Data memory read cycle timing .....                                      | 38 |
| 3.3 Penyambungan ROM .....   | 40 |
| 3.4 Rangkaian peralatan pendukung .....                                      | 41 |
| 3.5 Peta pemakaian port 1 .....  | 42 |
| 3.6 Rangkaian catu daya utama .....  | 45 |
| 3.7 Rangkaian catu daya darurat dan regulator tegangan<br>sistem utama ..... | 45 |
| 3.8 Rangkaian reset otomatis .....   | 46 |
| 3.9 Rangkaian detektor kegagalan catu daya .....                             | 46 |

# BAB I

## PENDAHULUAN

### 1.1 Latar belakang

Teknologi yang semakin canggih membawa pengaruh terhadap sistem pengamanan ruangan-ruangan yang bersifat pribadi atau terbatas, dimana untuk ruangan-ruangan tersebut perlu diberikan pengamanan yang dapat diandalkan.

### 1.2 Permasalahan

Sistem pengamanan yang umum digunakan adalah dengan menggunakan kunci pintu mekanik. Sistem ini hanya praktis untuk digunakan pada ruangan pribadi. Kelemahan sistem ini adalah pada modelnya yang sudah umum dimana mekaniknya mudah diperdaya. Sistem yang lain adalah sistem kunci elektronik yang tidak diperlengkapi dengan basis data. Sistem ini lebih bagus dari yang pertama karena bisa dirancang sangat rumit, pemegang kuncinya bisa lebih dari satu personil dan kodenya dapat dibuat berbeda untuk masing-masing kunci. Tetapi sistem ini masih tidak dapat mengenali siapa yang telah melewatinya, disamping itu juga tidak dapat memberikan

validasi terhadap kunci yang digunakan ataupun terhadap personil yang membawanya.

### 1.3 Maksud dan tujuan

Kunci elektronik yang akan dibahas di sini adalah kunci elektronik yang dapat mengatasi berbagai masalah di atas. Dengan adanya sistem basis data sebagai alat validasi dan pencatat, maka kunci elektronik ini menjadi alat pengaman ruangan yang dapat diandalkan.

Kunci elektronik ini terbagi atas dua bagian. Bagian pertama berupa sebuah kontroller yang dikendalikan oleh mikrokomputer 8031, yang menangani pembacaan kunci kartu, penerimaan password dan pembukaan pintu. Bagian kedua adalah sistem basis data yang menggunakan sebuah komputer IBM-PC yang berfungsi melakukan validasi dan pencatatan. Kedua bagian ini saling berhubungan melalui komunikasi serial.

### I.4 Sistematika pembahasan

Pembahasan dalam buku Tugas Akhir ini akan dilakukan dengan sistematika sebagai berikut :

- Bab dua membahas teori penunjang tentang mikrokomputer 8031 dan komunikasi serial pada komputer IBM-PC.
- Bab tiga membahas rancangan perangkat keras

kontroller.

- Bab empat membahas rancangan perangkat lunak yang digunakan untuk mengoperasikan kontroller.
- Bab lima membahas sistem basis data beserta rancangan perangkat lunaknya.
- Bab enam adalah kesimpulan dari tugas akhir ini.

## BAB II

### TEORI PENUNJANG

Bagian ini membahas teori penunjang yang dibutuhkan dalam perencanaan perangkat keras dari sistem kunci elektronik, yaitu mikrokomputer *chip* tunggal 8031 dan port komunikasi serial pada komputer IBM-PC.

#### 2.1 Mikrokomputer 8031<sup>1)</sup>

##### 2.1.1 Pendahuluan

8031 adalah suatu chip mikrokomputer buatan Intel yang dirancang khusus untuk digunakan sebagai sebuah mikrokontroler. Di dalam chip ini sudah terdapat :

- CPU
- 128 x 8 bit RAM
- 32 jalur I/O
- Dua timer / counter 16 bit
- Lima pelayanan interrupt dengan dua level prioritas
- Port serial I/O baik untuk komunikasi multiprosesor, perluasan I/O, atau *full duplex UART*
- Sebuah osilator dan clock.

---

1) Intel, *MCS-51 Family of Single Chip Microcomputers User's Manual*, Intel Corp. Santa Clara, USA, 1981, p. 1-1

ROM luar digunakan untuk menyimpan semua instruksi program. Jumlah total alamat ROM yang bisa diakses adalah 64K. Untuk stack dan tempat memanipulasi data disediakan RAM dalam (internal data memory) berkapasitas 256 byte. Bila dibutuhkan kapasitas yang lebih besar, maka RAM luar dapat dipasang dengan kapasitas sampai dengan 64k byte.

Stack dapat ditempatkan di mana saja di RAM dalam. RAM dalam juga merupakan tempat dari 21 register fungsi khusus, seperti *memory-mapped locations* dari port-port, register aritmatika, register kontrol dan status, serta timer/counter.

RAM dalam mempunyai 256 bit yang masing-masing dapat diakses sendiri. 128 bit terletak di register-register fungsi khusus dan 128 yang lain terletak di alamat lain. Bit-bit tersebut memberikan dimensi baru dalam aplikasi pemrogramannya, dimana programmer dapat memanipulasi masing-masing bit dengan instruksi bit. Kemampuan ini dimungkinkan oleh adanya suatu pengolah boolean atau pengolah bit yang mempunyai accumulator, I/O dan perintah-perintahnya sendiri.

Chip ini mempunyai dua buah timer/counter 16-bit yang berdiri sendiri. Dengan perintah perangkat lunak setiap timer/counter dapat dipasang pada salah satu dari empat mode berikut :

- 0> 8-bit timer/counter dengan suatu tingkatan pembagi 32
- 1> timer/counter 16 bit
- 2> timer/counter 8 bit *autoreload*
- 3> Suatu mode yang menghasilkan dua timer dan sebuah counter untuk penggunaan yang lebih fleksibel.

Komunikasi data serial diselenggarakan melalui saluran serial. Pengoperasiannya fleksibel dengan tersedianya baud rate yang dapat diprogram, dua pilihan ukuran frame dan kemampuan komunikasi multiprosesor. Empat mode operasi yang ada yaitu :

- 0> Frame 8-bit, synchronus mode yang digunakan untuk perluasan I/O menggunakan shift register luar yang di-clock oleh 8031.
- 1> Frame 8-bit, asynchronus mode dengan baud rate yang dapat diprogram dari 122 sampai 31250 bit per detik (Clock 12 MHz).
- 2> Frame 9-bit, asynchronus mode dengan baud rate 187,5 kbit per detik (Clock 12 MHz).
- 3> Asynchronus mode, baud rate dapat diprogram dari 122

sampai 31250 bit per detik dengan frame 9-bit.

Chip 8031 mempunyai empat buah port, port 0-3. Port 0 adalah sebuah *time multiplexed bus*. Port ini mengeluarkan 8 bit bagian bawah dari sistem alamat 16 bit dan juga mengirim / menerima data serta instruksi pada pengoperasian dengan menggunakan RAM / ROM luar. Port 2 mengeluarkan 8 bit bagian atas dari sistem alamat 16 bit untuk memori luar. Port 3 memuat hubungan-hubungan sinyal khusus seperti sinyal read dan write, dua input sinyal external interrupt, dua input counter serta saluran pengiriman dan penerimaan untuk komunikasi serial. Port 1 adalah port I/O 8 bit biasa.

Untuk menghemat pemakaian memori program, sebagian besar instruksi 8031 memiliki satu byte, yaitu 49 instruksi satu byte, 45 instruksi dua byte dan 17 instruksi tiga byte. Dengan menggunakan kristal 12 MHz sebagai sumber clock, 8031 memiliki 64 instruksi dengan waktu eksekusi 1  $\mu$ s, 45 instruksi dengan waktu eksekusi 2  $\mu$ s, dan instruksi perkalian serta pembagian hanya membutuhkan 4  $\mu$ s.

### 2.1.2 Fungsi-fungsi dalam 8031<sup>2)</sup>

Diagram blok fungsi dari 8031 diperlihatkan secara detil pada gambar 2.1. Fungsi dari tiap-tiap blok adalah sebagai berikut :

#### Dekoder Instruksi

Setiap instruksi program diterjemahkan oleh dekoder instruksi. Bagian ini akan membangkitkan sinyal yang mengontrol fungsi dari setiap bagian di dalam CPU.

#### Program Counter

Program Counter (PC) 16 bit berfungsi untuk mengontrol urutan instruksi yang akan dijalankan di dalam memori program.

#### RAM dalam

RAM dalam berkapasitas 256 byte. Pada 128 byte pertama terdapat :

- Bank Register

Terdapat empat buah Bank Register di dalam RAM data dalam, yang di setiap bank terdapat register R0 sampai R7.

- 128 Bit yang dapat dialamati

Terdapat dalam 16 byte yang berlokasi pada alamat 20H sampai 2FH di dalam RAM dalam.

---

2) Ibid., p. 2-2.

- **Stack**

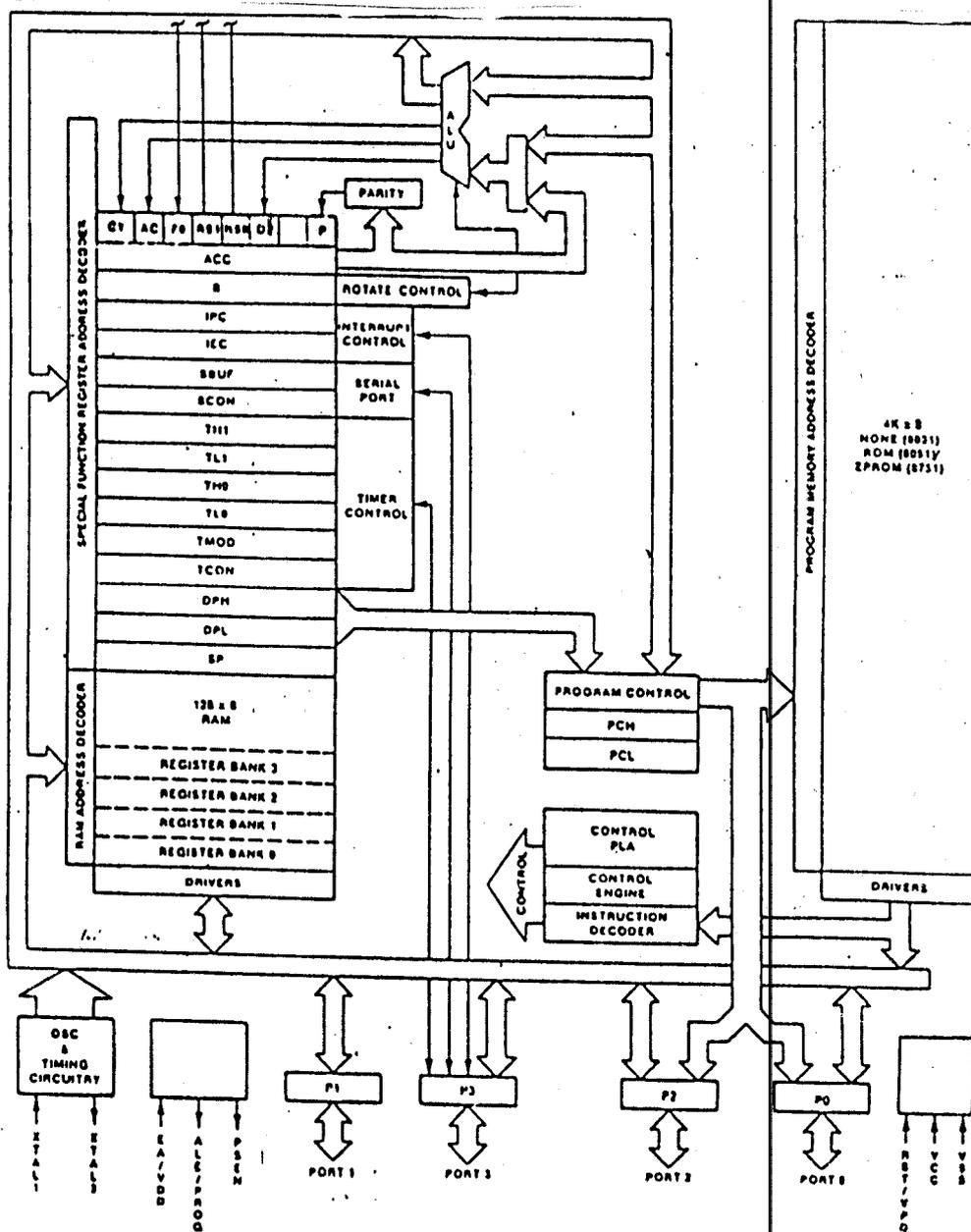
Stack dapat ditempatkan dimana saja pada RAM dalam. Kedalaman stack dapat mencapai 128 byte. Pada 128 byte yang lain dipakai sebagai tempat register fungsi khusus (SFR). Register fungsi khusus yang ditandai /\* adalah register-register yang dapat diakses baik secara byte maupun secara bit yaitu :

- **ACC /\***

Register A digunakan sebagai akumulator. Untuk menggunakan akumulator pada instruksi-instruksi yang menunjuk ke akumulator, register ini cukup disebut sebagai A, tetapi register itu sendiri dinamakan ACC.

- **Register B /\***

Register B digunakan bersama register A untuk instruksi perkalian dan pembagian. Pada instruksi-instruksi lain dapat diperlakukan sebagai register biasa.



Gambar 2.1 3)  
Diagram blok fungsi

3) Ibid., p.2-3.

- Register Program Status Word (PSW) /\*

Flag-flag Carry(CY), auxiliary carry (AC), user flag 0 (FO), pemilih bank register (RS0 dan RS1), overflow (OV) dan parity (P) terdapat dalam register ini. Flag CY, AC, dan OV pada umumnya menandakan keadaan dari operasi aritmatika yang terakhir. Flag P merupakan parity dari register A. Flag carry juga digunakan sebagai akumulator Boolean untuk operasi bit. FO adalah flag serba guna yang akan ikut di simpan ke stack sebagai bagian dari penyimpanan PSW.

- Stack Pointer

Stack pointer 8 bit menunjukkan alamat dari data terakhir yang disimpan ke stack pada pelaksanaan intruksi PUSH atau CALL. Isi stack pointer otomatis dinaikkan pada pelaksanaan instruksi PUSH dan diturunkan pada pelaksanaan instruksi POP.

- DPTR

Data Pointer (DPTR) adalah register 16-bit, yang terdiri dari byte bagian atas (DPH) dan byte bagian bawah (DPL). Fungsinya adalah untuk menyimpan alamat 16-bit sebagai pointer ke ROM dan RAM luar.

- P0, P1, P2 dan P3 /\*

Empat buah port yang menghasilkan 32 jalur I/O untuk berhubungan dengan peralatan luar. Tiap port terdiri dari latch (register fungsi khusus P0, P1, P2 dan

P3), *driver* keluaran dan *buffer*. masukan. Driver keluaran dari port 0 dan port 2, serta buffer masukan dari port 0 digunakan dalam mengakses memori luar. Dalam aplikasi ini, port 0 digunakan untuk mengeluarkan byte bagian bawah dari alamat memori luar yang dimultipleks dengan byte yang sedang ditulis atau dibaca. Sedang Port 2 (P2) dapat digunakan untuk mengeluarkan byte bagian atas dari alamat memori luar. Port 1 adalah port I/O serba guna sedangkan port 3 adalah port khusus yang digunakan baik untuk komunikasi, external interrupt maupun untuk perluasan I/O.

- SBUF

*Serial Data Buffer* (SBUF) terdiri dari dua register. Jika data dikirimkan ke SBUF, ia akan menuju ke *transmit buffer* dimana diperlukan untuk transmisi serial. Apabila data diambil dari SBUF, ia akan berasal dari *receive buffer*.

Selama berlangsungnya penerimaan, bit-bit serial yang datang di-clock-kan ke shift register khusus. Apabila penerimaan data sudah lengkap dan berbagai kondisi sudah terpenuhi, data 8 bit yang diterima dipindahkan dari shift register ke buffer penerima. Selanjutnya shift register siap menerima frame berikutnya.

- IP /\*

Register Prioritas Interrupt (IPC) berisi bit-bit kontrol untuk memasang interrupt pada tingkat prioritas yang diinginkan.

- IEC /\*

Register Enable Interrupt (IEC) digunakan untuk menyimpan bit-bit *enable* dari kelima sumber interrupt dan untuk menyimpan bit *enable/disable* dari keseluruhan sistem interrupt.

- TMOD

Register mode timer/counter. Bit-bit pada register Timer Mode (TMOD) digunakan untuk memilih timer/counter yang akan bekerja.

- TCON /\*

Semua timer/counter dikendalikan oleh bit-bit di dalam register ini. Bit-bit start/stop untuk semua timer/Counter, flag-flag overflow dan interrupt request disimpan dalam TCON.

- TH1 Timer/Counter 1 (High), TL1 Timer/Counter 1 (Low), TH0 Timer/Counter 0 (High) dan TL0 Timer/Counter 0 (Low)

Terdapat empat lokasi register untuk dua buah Timer/Counter 16 bit. Register-register ini dapat dibaca dan ditulisi. TH1 dan TH0 adalah 8 bit bagian

atas dari timer/counter 1 dan 0. Sedangkan TL1 dan TLO adalah 8 bit bagian bawah dari timer/counter 1 dan 0.

- **SCON /\***

Serial Control Register (SCON) memuat bit-bit untuk mengaktifkan penerimaan port serial. Pemilihan mode operasi dari port serial juga dilakukan dengan bit-bit pada register tersebut.

- **SBUF**

Register Serial Data Buffer (SBUF) digunakan untuk menampung data masukan atau keluaran dari port.

- **Bagian aritmatika**

Bagian aritmatika dari prosesor membentuk beberapa fungsi manipulasi data yang dilaksanakan oleh *Arithmetic Logic Unit* (ALU), dengan menggunakan register A, register B dan register PSW. ALU membentuk operasi-operasi aritmatika seperti penjumlahan, pengurangan, perkalian, pembagian, *increment*, *decrement*, perbandingan, *BCD-decimal-add-adjust*, dan operasi logika seperti AND, OR, EXOR, pembalikan dan pemutaran (ke kanan, ke kiri) serta pertukaran *nibble* (empat bit kiri dan kanan ditukar).

#### Rangkaian oscillator dan timing<sup>4)</sup>

Rangkaian oscillator yang terdapat didalam Chip adalah suatu rangkaian "*Parallel Anti Resonant*" dengan batas frekuensi mulai dari 1,2MHz sampai 12MHz. Frekuensi tersebut dibagi 12 sehingga menghasilkan siklus instruksi minimum 1μ detik bila menggunakan kristal 12MHz. Kaki XTAL2 adalah keluaran dari amplifier penguatan tinggi, sedangkan kaki XTAL1 sebagai masukannya. Kristal dihubungkan dengan kaki XTAL1 dan XTAL2 menghasilkan umpan balik dan penggeseran fasa yang dibutuhkan untuk beresilasi. Jika XTAL1 dihubungkan dengan sumber frekuensi dari luar, XTAL2 tidak perlu dihubungkan ke kristal.

#### Penroses Boolean<sup>5)</sup>

Penroses Boolean adalah suatu pemroses bit yang berdiri sendiri. Memiliki instruksi-instruksi, akumulator (register carry), RAM dalam dengan bit-bit yang dapat dialamati dan I/O port sendiri. Bit-bit tersebut terdapat pada RAM dalam sebanyak 128 bit dan pada register-register fungsi khusus (SFR) sebanyak 128 bit. Register-register P0, P1, P2, P3, TCON,

4) *Ibid.*, p.2-6.

5) *Ibid.*, p.2-4.

SCON, IEC, IPC, PSW, A dan B pada SFR mempunyai bit-bit yang dapat diakses secara langsung.

Pemroses Boolean dapat juga mengerjakan operasi-operasi bit seperti set, clear, komplemen, *jump-if-set*, *jump-if-not -set*, *jump-if-set-then-clear* dan pemindahan dari atau ke akumulatornya (flag Carry).

### 2.1.3 Sistem interrupt<sup>6)</sup>

Mikrokomputer 8031 mempunyai lima sumber interrupt, yang dapat diprogram pada salah satu dari dua tingkat prioritas. Lima sumber interrupt itu adalah :

- INT 0 : permintaan dari luar (*external request*) dari kaki P3.2 (port 3 bit 2).
- INT 1 : permintaan dari luar dari kaki P3.3
- timer 0 : Overflow dari timer 0 akan mengaktifkan flag permintaan interrupt (*interrupt request flag*) TFO.
- timer 1 : Overflow dari timer 1 akan mengaktifkan flag permintaan interrupt TF1.
- Port Serial :  
Flag TI akan aktif bila pengiriman satu frame serial telah lengkap dan flag RI

---

<sup>6)</sup>*Ibid.*, p.2-5.

akan aktif bila penerimaan satu frame serial telah lengkap.

Alamat-alamat awal dari rutin program pelayanan interrupt, dimana Program Counter akan melompat pada alamat tersebut pada saat ada permintaan interupsi, dapat dilihat pada Tabel 2.1.

Setiap sumber interrupt dapat di-enable atau di-disable secara tersendiri dengan mengubah isi bit pada register Interrupt Enable (IE) menjadi 1 atau 0. Semua sumber interrupt juga dapat di-enable atau di-disable sekaligus.

TABEL 2.1<sup>7)</sup>  
ALAMAT AWAL DARI PROGRAM PELAYANAN INTERRUPT

| Sumber Interrupt        | Alamat awal |
|-------------------------|-------------|
| Interrupt Luar 0 (INT0) | 3 (0003 H)  |
| timer/counter 0 (T0)    | 11 (000B H) |
| Interrupt Luar 1 (INT1) | 19 (0013 H) |
| timer/counter 1 (T1)    | 27 (001B H) |
| Port Serial             | 35 (0023 H) |

7) Ibid., p.2-6.

|       |   |   |    |     |     |       |     |
|-------|---|---|----|-----|-----|-------|-----|
| (MSB) |   |   |    |     |     | (LSB) |     |
| 7     | 6 | 5 | 4  | 3   | 2   | 1     | 0   |
| EA    | X | X | ES | ET1 | EX1 | ETO   | EXO |

GAMBAR 2.2<sup>8)</sup>  
REGISTER INTERRUPT ENABLE

|       |   |   |    |     |     |       |     |
|-------|---|---|----|-----|-----|-------|-----|
| (MSB) |   |   |    |     |     | (LSB) |     |
| 7     | 6 | 5 | 4  | 3   | 2   | 1     | 0   |
| X     | X | X | PS | PT1 | PX1 | PTO   | PXO |

GAMBAR 2.3<sup>9)</sup>  
REGISTER PRIORITAS INTERRUPT

Prioritas interrupt dipilih dengan mengubah isi bit-bit dari register prioritas interrupt (IP). Susunan bit-bit dari register EI digambarkan pada Gambar 2.2.

Posisi bit-bit pada register, nama dan artinya ditunjukkan pada Tabel 2.2. Sedangkan untuk register IP ditunjukkan pada Gambar 2.3 dan Tabel 2.3. Suatu rutin

---

8) *Ibid.*, p.B-13.

9) *Ibid.*, p.B-13.

TABEL 2.2<sup>10)</sup> NAMA DAN ARTI DARI BIT-BIT  
PADA REGISTER INTERRUPT ENABLE

| SIMBOL | POSISI | NAMA DAN ARTI   |
|--------|--------|---|
| EA     | IE.7   | Enable All. Bila EA = 0, semua interrupt di-disable, tidak tergantung keadaan IEO - IE.4. |
| ES     | IE.4   | Enable Serial port. Bila ES = 0, interrupt port serial disable.                           |
| ET1    | IE.3   | Enable Timer 1 . Bila ET1 = 0, interrupt timer 1 disable.                                 |
| EX1    | IE.2   | Enable Interrupt luar 1. Bila EX1 = 0, interrupt luar 1 disable.                          |
| ETO    | IE.1   | Enable Timer 0. Bila ETO = 0, interrupt timer 0 disable.                                  |
| EXO    | IE.0   | Enable interrupt luar 0. Bila EXO = 0, interrupt luar 0 disable.                          |

pelayanan interrupt dapat diinterupsi oleh interrupt lain dengan prioritas yang lebih tinggi, tetapi tidak dapat diinterupsi oleh interrupt dengan prioritas yang lebih rendah atau sama.

Register TCON digunakan untuk menentukan pemicu interrupt luar pada pengaktifan pinggir negatif (falling

10) *Ibid.*, p. B-13.

edge) atau logika rendah (low level), menjalankan atau menghentikan timer/counter. Timer Overflow Flag pada register ini akan di-set oleh perangkat keras jika terjadi overflow pada timer/counter. Susunan bit-bit dari register TCON digambarkan pada Gambar 2.4. Posisi bit-bit pada register, sebutan dan artinya ditunjukkan pada Tabel 2.3.

#### 2.1.4 Timer/counter<sup>11)</sup>

8031 mempunyai empat buah register yang khusus digunakan oleh timer/counter. Register-register tersebut adalah TH0 (8 bit bagian atas dari timer/counter 0) dan TLO (8 bit bagian bawah dari timer/counter 0) untuk timer/counter 0, TH1 (8 bit bagian atas dari timer/counter 1) dan TL1 (8 bit bagian bawah timer/counter 1) untuk timer/counter 1. Setiap timer/counter dikontrol oleh bit dalam register TMOD untuk memilih fungsinya sebagai timer atau counter.

---

11) *Ibid.*, p.2-11.

TABEL 2.3<sup>12)</sup>  
 NAMA DAN ARTI DARI BIT-BIT  
 PADA REGISTER INTERRUPT PRIORITY

| SIMBOL | POSISI | NAMA DAN ARTI  |
|--------|--------|--|
| PS     | IP.4   | Tingkat prioritas serial port.<br>Bila PS = 1 tingkat prioritasnya menjadi tinggi.       |
| PT1    | IP.3   | Tingkat prioritas timer 1.<br>Bila PT1 = 1 tingkat prioritasnya menjadi tinggi.          |
| PX1    | IP.2   | Tingkat prioritas interrupt luar 1.<br>Bila PX1 = 1 tingkat prioritasnya menjadi tinggi. |
| PTO    | IP.1   | Tingkat prioritas timer 0.<br>Bila PTO = 1 tingkat prioritasnya menjadi tinggi.          |
| PX0    | IP.0   | Tingkat prioritas interrupt luar 0.<br>Bila PX1 = 1 tingkat prioritasnya menjadi tinggi. |

| (MSB) |     |     |     | (LSB) |     |     |     |
|-------|-----|-----|-----|-------|-----|-----|-----|
| 7     | 6   | 5   | 4   | 3     | 2   | 1   | 0   |
| TF1   | TR1 | TFO | TRO | IE1   | IT1 | IE0 | IT0 |

GAMBAR 2.4<sup>13)</sup>  
 REGISTER TIMER/COUNTER CONTROL/STATUS (TCON)

12) *Ibid.*, p. B-13.

13) *Ibid.*, p.B-11.

TABEL 2.4  
 NAMA DAN ARTI DARI BIT-BIT  
 PADA REGISTER TCON<sup>14)</sup>

| SIMBOL | POSISI | NAMA DAN ARTI   |
|--------|--------|---|
| TF1    | TCON.7 | Flag overflow timer 1. Set oleh perangkat keras bila timer/counter overflow. Clear bila interrupt diproses.         |
| TR1    | TCON.6 | Bit kontrol timer 1. Set / clear oleh perangkat lunak untuk menjalankan/menghentikan timer/counter.                 |
| TFO    | TCON.5 | Flag overflow timer 0. Set oleh perangkat keras bila timer/counter overflow. Clear bila interrupt diproses.         |
| TRO    | TCON.4 | Bit kontrol timer 0. Set / clear oleh perangkat lunak untuk menjalankan/menghentikan timer/counter.                 |
| IE1    | TCON.3 | Flag Interrupt luar 1. Set oleh perangkat keras bila sisi interrupt luar terdeteksi. Clear bila interrupt diproses. |

14) *Ibid.*, p.8-11.

TABEL 2.4  
 NAMA DAN ARTI DARI BIT-BIT  
 PADA REGISTER TCON (Lanjutan)

| SIMBOL | POSISI | NAMA DAN ARTI   |
|--------|--------|---|
| IT1    | TCON.2 | Bit kontrol interrupt luar 1. Set / clear oleh perangkat lunak untuk menentukan cara pengaktifan transisi atau level rendah pada masukan dari interrupt luar 1. |
| IE0    | TCON.1 | Flag interrupt luar 1. Set oleh perangkat keras bila sisi interrupt luar terdeteksi. Clear bila interrupt diproses.   |
| IT0    | TCON.0 | Bit kontrol interrupt luar 0. Set / clear oleh perangkat lunak untuk menentukan cara pengaktifan transisi atau level rendah pada masukan dari interrupt luar 0. |

Timer/counter dapat dioperasikan dalam empat mode yang diinginkan, yaitu:

- Mode 0

Mengoperasikan counter/timer 8 bit dengan tingkatan pembagi 32. Register TH1 atau TH0 sebagai timer/ counter, register TL1 atau TLO bit 4-0 sebagai tingkatan, sedangkan bit 7-5 diabaikan.

- Mode 1

Mengoperasikan counter 1 sebagai timer/counter 16 bit.

- Mode 2

Mengoperasikan counter 1 sebagai timer/counter 8 bit autoreload. Register TH1 berisi bilangan yang akan di-reload ke register TL1 setiap kali overflow terjadi.

- Mode 3

Untuk counter/timer 1, perubahan pemrograman dari mode 0-2 menjadi mode 3 menghentikan counter. Ini adalah suatu cara alternatif selain menggunakan TR1.

Pada timer/counter 0, counter/timer 1 dihentikan karena TR1 digunakan untuk mengontrol kenaikan isi TH0 yang sebuah timer/counter 8 bit yang terpisah dari TLO yang juga menjadi sebuah timer/counter 8 bit sendiri. Ketika TH0 mengalami overflow, TF1 di-set. TH0 tidak dipengaruhi oleh keadaan bit-bit dalam TCON atau TMOD. TLO tetap dikontrol oleh TR0, TFO dan C/T (TMOD.2).

Susunan bit-bit dari register TMOD digambarkan pada Gambar 2.5.

| (MSB) |              |    |    | (LSB) |              |    |    |
|-------|--------------|----|----|-------|--------------|----|----|
| 7     | 6            | 5  | 4  | 3     | 2            | 1  | 0  |
| GATE  | C/ $\bar{T}$ | M1 | M0 | GATE  | C/ $\bar{T}$ | M1 | M0 |

GAMBAR 2.5  
REGISTER TMOD (TIMER/COUNTER MODE)<sup>15)</sup>

Fungsi dari tiap-tiap bit adalah sebagai berikut :

- GATE

Kontrol gerbang. Bila GATE = 1, timer/counter 0 atau 1 akan di-enable hanya bila kaki  $\overline{INT0}/\overline{INT1}$  dan TO/T1 mendapat masukan tinggi dan bit kontrol TRO/TR1 = 1. Apabila GATE = 0, timer 0/timer 1 di-enable ketika TO/T1 mendapat masukan tinggi dan TRO/TR1 = 1.

- C/ $\bar{T}$

Pemilih operasi timer atau counter. C/ $\bar{T}$  = 0 untuk operasi timer (masukan didapat dari sistem clock di dalam), dan C/ $\bar{T}$  = 1 untuk operasi counter (masukan dari kaki TO/T1).

<sup>15)</sup> *Ibid.*, p.B-11.

- M1 dan M0

Pemilih mode operasi. M1 = 0 dan M0 = 0 untuk mode 0, M1 = 0 dan M0 = 1 untuk mode 1, M1 = 1 dan M0 = 0 untuk mode 2, M1 = 1 dan M0 = 1 untuk mode 3.

### 2.1.5 Port serial<sup>16)</sup>

Salah satu keunggulan dari mikrokomputer 8031 adalah dengan adanya port serial yang menjadi satu di dalam chip. Port serial ini dapat digunakan untuk komunikasi data serial, baik yang menggunakan hubungan half duplex maupun full duplex, atau untuk memperluas port I/O dengan menghubungkan ke shift register.

Port serial ini dapat dioperasikan ke dalam empat mode, yaitu :

- Mode 0 (synchronous)

Data serial 8 bit dikirim dan diterima melalui RXD, dengan bit terendah (LSB) lebih dahulu, dan TXD mengeluarkan clock penggeser (*shift clock*). Baud rate ditetapkan (fix) pada 1/12 frekuensi osilator.

---

<sup>16)</sup> *Ibid.*, p.2-13.

- **Mode 1 (asynchronous)**

10 bit dikirimkan melalui TXD atau diterima melalui RXD dengan urutan : satu start bit, 8 bit data (LSB yang pertama) dan satu stop bit. Pada penerimaan data, stop bit akan mengisi RB8 pada register SCON. Baud rate dapat diubah-ubah (variabel).

- **Mode 2 (asynchronous)**

11 bit dikirimkan melalui TXD atau diterima melalui TXD dengan urutan : satu start bit, 8 bit data (LSB yang pertama), satu bit data yang dapat diprogram (bit ke sembilan), dan satu stop bit. Pada pengiriman data, bit data ke sembilan (TB8) dapat dipilih '0' atau '1'. TB8 dapat digunakan sebagai parity bit. Pada penerimaan data, bit data kesembilan akan mengisi RB8 pada register SCON, dan stop bit diabaikan. Baud rate ditetapkan pada 1/64 dari frekuensi osilator.

- **Mode 3 (asynchronous)**

Mode 3 sama dengan Mode 2 kecuali pada Mode 3 baud rate dapat diubah-ubah (variabel).

### 2.1.5.1 Baud rate <sup>17)</sup>

Baud rate untuk mode 0 adalah tetap pada 1/12 dari frekuensi osilator. Untuk mode 2 baud rate adalah 1/64. Baud rate pada mode 1 dan 3 ditentukan oleh overflow rate dari Timer 1, dengan persamaan sbb. :

$$\text{Baud rate} = (\text{overflow rate Timer 1}) / 32$$

Overflow rate Timer 1 ditentukan oleh kecepatan pencacahan (count rate = 1/12 f osilator) dan banyaknya pencacahan yang diperlukan untuk mencapai overflow.

Persamaan untuk overflow rate adalah :

$$\text{Overflow rate} = (f_{\text{osc}} \text{ rate}) / [256 - (TH1)]$$

(TH1) adalah nilai reload yang diisikan dengan perangkat lunak pada register TH1.

|          |          |          |          |          |          |         |         |
|----------|----------|----------|----------|----------|----------|---------|---------|
| (MSB)    |          |          |          |          |          | (LSB)   |         |
| 7<br>SM0 | 6<br>SM1 | 5<br>SM2 | 4<br>REN | 3<br>TB8 | 2<br>RB8 | 1<br>TI | 0<br>RI |

GAMBAR 2.6  
REGISTER SCON

17) *ibid.*, p.2-16.

### 2.1.5.2 Register kontrol port serial (SCON)<sup>18)</sup>

Register SCON digunakan untuk mendefinisikan mode operasi dan kontrol fungsi-fungsi tertentu dari port serial. Juga menerima bit data ke 9 (RB8) dan interrupt flag dari pengiriman dan penerimaan (TI dan RI). Register SCON digambarkan pada Gambar 2.6 seperti di bawah ini.

SM0 dan SM1 digunakan untuk memilih mode operasi dari port serial seperti yang ditunjukkan pada Tabel 2.4

TABEL 2.5<sup>19)</sup>  
PEMILIHAN MODE OPERASI DARI PORT SERIAL  
DENGAN MENGGUNAKAN BIT-BIT SM0 DAN SM1

| SM0 | SM1 | MODE | FUNGSI         | BAUD RATE      |
|-----|-----|------|----------------|----------------|
| 0   | 0   | 0    | shift register | $f_{osc} / 12$ |
| 0   | 1   | 1    | 8 bit UART     | variabel       |
| 1   | 0   | 2    | 9 bit UART     | $f_{osc} / 64$ |
| 1   | 1   | 3    | 9 bit UART     | variabel       |

Sedangkan fungsi bit-bit yang lain adalah :

#### - SM2

Pada Mode 2 atau 3, bila SM2 = 1, maka RI tidak akan diaktifkan bila bit data ke 9 (RB8) yang diterima adalah "0". Pada Mode 1, jika SM2 = 1, maka RI tidak akan diaktifkan bila stop bit yang

18) *Ibid.*, p.B-12.

19) *Ibid.*, p.C-6.

sempurna tidak diterima. Pada Mode 0 SM2 tidak digunakan.

- REN

Set / clear oleh perangkat lunak untuk enable / disable penerimaan data.

- TB8

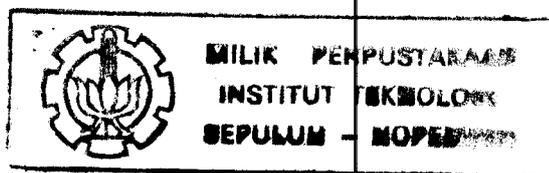
Bit data ke 9 yang akan dikirimkan pada Mode 2 dan 3.

- RB8

Pada Mode 2 dan 3, RB8 sebagai bit data ke 9 yang diterima. Pada Mode 1, bila SM2 = 0, RB8 adalah stop bit yang diterima. Pada Mode 0 RB8 tidak digunakan.

- TI

Sebagai *transmit interrupt flag* yang di-set oleh perangkat keras pada akhir dari waktu bit ke 8 pada Mode 0, atau pada permulaan dari stop bit pada mode yang lain, dalam setiap pengiriman serial. Clear TI hanya oleh perangkat lunak.



- RI

Sebagai *receive interrupt flag* yang di-set oleh perangkat keras pada akhir dari waktu bit ke 8 pada Mode 0, atau separuh dari pemasukan stop bit pada mode yang lain, dalam setiap penerimaan serial. Clear RI hanya oleh perangkat lunak.

## 2.2. KOMUNIKASI SERIAL ASINKRON

### 2.2.1 Standard Interface RS-232C

RS-232C adalah interface elektrik standard untuk menghubungkan komponen - komponen sistem seperti Modem, Printer dan Komputer. Standard ini pada tahun 1969 ditetapkan oleh EIA (*Electronic Industries Association*) sebuah organisasi perdagangan industri, kemudian diperluas melalui CCITT (*Consultative Comitte on International Telegraphy and Telephony*) salah satu komite dari ITU (*International Telecommunication Union*) yaitu salah satu lembaga di PBB yang membidangi masalah telekomunikasi dunia.

RS-232C menetapkan 25-jalur sinyal yang membentuk 18 rangkaian dengan jalur yang kembali lewat ground. Standard juga menetapkan range tegangan untuk logika "0" dan logika "1", level impedansi, waktu naik/jatuh, laju bit maksimum, dan kapasitansi maksimum yang digunakan

dalam semua rangkaian.

Tabel 2.6 dibawah menunjukkan nama sinyal, arah sinyal dan uraian fungsi masing-masing pin RS-232C. Pin-1 Protective Ground secara tak resmi disebut *Chasis Ground*. Jika suatu peralatan tidak mempunyai Ground pada plug AC power supply-nya, maka peralatan tersebut perlu untuk dihubungkan ke pin-1, ini untuk mencegah kejutan listrik.

Fungsi pin ini seringkali dirancukan dengan pin-7 *Common Return*, Cabang tengah cord AC pada akhirnya harus kembali ke pentanahan induk, karena itu diistilahkan *Ground*. Dalam prakteknya secara tidak langsung jalur ini juga dapat menghasilkan resistansi antara peralatan dan bumi. Saat dua peralatan dihubungkan ke jaringan distribusi yang berbeda, maka jalur pembumian keduanya bisa berbeda. Secara elektronik Ini akan menyebabkan casis keduanya tidak sama, walaupun kondisi ini dapat mengganggu komunikasi, tetapi masih dapat diatasi dengan menghubungkan ke dua casis lewat pin-1.

Beberapa pin lain yang digunakan dalam interface mikro komputer sebagian besar hanya menggunakan sembilan pin. sekumpulan pin 2,3,4,5,6,7,8,20 yang sering dipakai disebut *Delapan Besar*.

TABEL 2.6 <sup>20)</sup>  
NAMA PIN DAN URAIAN SINYAL RS-232C

| Pin | Common Name             | RS-232C Name | Description                                   | Direction on DCE |
|-----|-------------------------|--------------|---|------------------|
| 1   |                         | AA           | Protective Ground                             | —                |
| 2   | TXD                     | BA           | Transmitted Data                              | IN               |
| 3   | RXD                     | BB           | Received Data                                 | OUT              |
| 4   | $\overline{\text{RTS}}$ | CA           | Request to Send                               | IN               |
| 5   | $\overline{\text{CTS}}$ | CB           | Clear to Send                                 | OUT              |
| 6   | $\overline{\text{DSR}}$ | CC           | Data Set Ready                                | OUT              |
| 7   | GND                     | AB           | Signal Ground (Common Return)                 | —                |
| 8   | $\overline{\text{CD}}$  | CF           | Received Line Signal Detector                 | OUT              |
| 9   |                         | —            | (Reserved for Data Set Testing)               | —                |
| 10  |                         | —            | (Reserved for Data Set Testing)               | —                |
| 11  |                         |              | Unassigned                                    | —                |
| 12  |                         | SCF          | Secondary Rec'd. Line Sig. Detect             | OUT              |
| 13  |                         | SCB          | Secondary Clear to Send                       | OUT              |
| 14  |                         | SBA          | Secondary Transmitted Data                    | IN               |
| 15  |                         | DB           | Transmission Sig. Element Timing (DCE Source) | OUT              |
| 16  |                         | SBB          | Secondary Received Data                       | OUT              |
| 17  |                         | DD           | Receiver Signal Element Timing (DCE Source)   | OUT              |
| 18  |                         |              | Unassigned                                    | —                |
| 19  |                         | SCA          | Secondary Request to Send                     | IN               |
| 20  | $\overline{\text{DTR}}$ | CD           | Data Terminal Ready                           | IN               |
| 21  |                         | CG           | Signal Quality Detector                       | OUT              |
| 22  |                         | CE           | Ring Indicator                                | OUT              |
| 23  |                         | CH/CI        | Data Signal Rate Selector (DTE/DCE Source)    | IN/OUT           |
| 24  |                         | DA           | Transmit Signal Element Timing (DTE Source)   | IN               |
| 25  |                         |              | Unassigned                                    | —                |

20) V. Hall, Douglas, *Microprocessor and Interfacing : Programming and Hardware*, McGraw Hill, 1986, p. 451

### 2.2.2 Adapter Komunikasi Serial Asinkron IBM-PC

Adapter Komunikasi Asinkron menyediakan sinyal kontrol dan tegangan yang diperlukan lewat tab 2X31 pada sisi card. Dua modul jumper disediakan pada adapter. Satu modul jumper digunakan untuk memilih operasi RS-232C atau *Current-Loop*. Jumper yang lain digunakan untuk memilih salah satu dari dua alamat adapter, jadi dua adapter dapat digunakan dalam satu sistem.

Adapter secara penuh dapat diprogram dan hanya mendukung komunikasi asinkron. Adapter juga akan menyisipkan dan menghapus start-bits, stop-bits, dan parity-bits. Generator Baud Rate terprogram memungkinkan beroperasi dari 50 baud sampai 9600 baud. lima, enam, tujuh, atau delapan bit karakter dengan 1, 1 1/2, atau 2 stop-bit telah disediakan.

Servis 14h/0 memberikan dua harga output yang menunjukkan status dan kondisi adapter komunikasi asinkron, dimana informasi tersebut tersimpan pada register AH untuk status line dan pada register AL untuk status modem, demikian juga untuk servis 3.

Arti tiap bit dari status line dan status modem ditunjukkan pada Tabel 2.8 dan Tabel 2.9.

TABEL 2.7 21)  
ALAMAT ADAPTER KOMUNIKASI

| Primary Adapter | Alternate Adapter | Register Selected             | DLAB state |
|-----------------|-------------------|-------------------------------|------------|
| 3F8             | 2F8               | TX Buffer                     | DLAB=0     |
| 3F8             | 2F8               | RX Buffer                     | DLAB=0     |
| 3F8             | 2F8               | Divisor Latch LSB             | DLAB=1     |
| 3F9             | 2F9               | Divisor Latch MSB             | DLAB=1     |
| 3F9             | 2F9               | Interrupt Enable Register     |            |
| 3FA             | 2FA               | Interrupt Identification Reg. |            |
| 3FB             | 2FB               | Line Control Register         |            |
| 3FC             | 2FC               | Modem Control Register        |            |
| 3FD             | 2FD               | Line Status Register          |            |
| 3FE             | 2FD               | Modem Status Register         |            |

TABEL 2.8 22)  
ARTI TIAP BIT DARI AL UNTUK SERVIS 14h/0

| 76543210 | ARTI              |
|----------|-------------------|
| 000      | 110 baud          |
| 001      | 150 baud          |
| 010      | 300 baud          |
| 011      | 600 baud          |
| 100      | 1200 baud         |
| 101      | 2400 baud         |
| 110      | 4800 baud         |
| 111      | 9600 baud         |
| 00       | No parity         |
| 01       | Odd parity        |
| 10       | No parity         |
| 11       | Even parity       |
| 0        | 1 stop bit        |
| 1        | 2 stop bits       |
| 10       | 7-bit data length |
| 11       | 8-bit data length |

21) IBM, *Personal Computer XT System Technical Reference*, IBM Corporation, USA, p. 1-87

22) Wyatt, Allen L., *Using Assembly Language*, -, p 307.

TABEL 2.9<sup>23)</sup>  
 ARTI TIAP BIT DARI STATUS LINE REGISTER

| bit<br>76543210 | Meaning                 |
|-----------------|-------------------------|
| 1               | Time-out error          |
| 1               | TSR emty                |
| 1               | THR emty                |
| 1               | Break interupt detected |
| 1               | Framing error           |
| 1               | Parity error            |
| 1               | Overun error            |
| 1               | Data ready              |

TABEL 2-11<sup>24)</sup>  
 ARTI TIAP BIT DARI STATUS MODEM REGISTER

| bit<br>76543210 | Meaning                           |
|-----------------|-----------------------------------|
| 1               | Sinyal receiver line              |
| 1               | Ring indikator                    |
| 1               | DSR                               |
| 1               | CTS                               |
| 1               | Delta receiver line signal detect |
| 1               | Trailing edge ring detector       |
| 1               | DDSR                              |
| 1               | DCTS                              |

23) *Ibid.*, p. 308

24) Wyatt, Allen L., *loc.cit*

## BAB III

### RANCANGAN PERANGKAT KERAS

#### 3.1 Konfigurasi

Mikrokomputer 8031 dirancang untuk menggunakan ROM luar untuk menyimpan perangkat lunaknya . Oleh karena itu port 0 akan digunakan sebagai BUS alamat dan data yang dihubungkan dengan ROM luar. Port ini dihubungkan pada latch alamat karena penggunaannya bergantian dengan BUS data.

Pada gambar 3.1 dan 3.2 dapat dilihat diagram waktu untuk pembacaan dari memori program dan dari memori data.

Port 2 digunakan sebagai BUS untuk byte bagian atas dari word alamat. Rancangan ini memakai ROM berkapasitas 2kB, sehingga port 2 akan terpakai 3 buah pinnya sebagai BUS alamat ROM luar. Lihat gambar 3.3

Port 0 juga digunakan untuk menerima data dari peralatan-peralatan pendukung. Untuk itu port ini dihubungkan dengan keluaran *Three State Buffer* 8 bit yang dikendalikan oleh pin  $\overline{RD}$ .

## BAB IV

### RANCANGAN PERANGKAT LUNAK

#### IV.1 Pendahuluan

Perangkat lunak kontroller disusun dalam bahasa program assembler MCS-51. Di dalamnya terdapat 51 buah instruksi dasar yang dibagi lagi menjadi lima kelompok fungsi. Ditambah lagi dengan beberapa mode pengalamatan yaitu jenis data boolean (1 bit), nibble (4 bit), byte (8 bit) dan alamat (16 bit) menghasilkan 111 buah instruksi.

Setiap instruksi program mengandung sebuah kata-kata operasi dan sampai empat buah *operand* (tergantung pada instruksinya). Instruksi yang memerlukan dua data operand selalu menjadikan yang pertama sebagai penerima hasil operasi, kecuali bila kedua operand dalam mode *direct*. Dalam kasus ini operand kedualah yang menjadi penerima hasil operasi.

Memori diorganisasikan menjadi tiga ruang alamat dan sebuah program counter. Ruang memori dapat dilihat pada gambar 4.1. Ruang-ruang tersebut adalah sbb.:

- 16 bit program counter
- 64K-byte total ruang alamat memori program

- 64K-byte ruang alamat memori data luar
- 384-byte ruang alamat memori data dalam

Register Program Counter 16-bit memungkinkan pengalamatan dengan kemampuan memori 64K. Program Counter memungkinkan pelaksanaan instruksi pemanggilan subrutin dan pencabangan ke semua lokasi di dalam ruang program memori. Tidak ada instruksi yang memungkinkan pelaksanaan program untuk berpindah dari ruang memori program ke suatu ruang memori data.

Beberapa lokasi di dalam memori program disediakan untuk dipakai oleh program khusus. Lokasi 0000 sampai 0002 dipakai untuk program inisialisasi. Setelah terjadi *reset*, CPU selalu memulai pelaksanaan program pada lokasi 0000. Lokasi 0003 sampai 0042 disediakan untuk program-program pelayanan permintaan interrupt.

64 kB ruang memori data luar diakses dengan instruksi MOVX. Memori data luar di dalam rancangan ini tidak dipakai. Tetapi instruksi MOVX akan dipakai untuk membaca data dari peralatan-peralatan pendukung yang diperlakukan seolah-olah sebagai memori data luar.

## IV.2 Program inisialisasi

Pada saat inisialisasi, kondisi-kondisi dari perangkat keras dipersiapkan agar dapat beroperasi seperti yang diinginkan. Kondisi-kondisi tersebut meliputi pemasangan mode pengaktifan interrupt luar IE0 (untuk mendeteksi kegagalan catu daya), pemasangan mode pengoperasian timer/counter dan saluran serial untuk komunikasi serial, serta penentuan bank register yang akan dipakai.

RAM data dalam di alamat-alamat 0000 sampai 0007 digunakan untuk menyimpan data sementara dari kunci kartu dan tombol password. Alamat 0020h sampai 0026 dipakai untuk menyimpan data kartu dan password khusus.

```

;alamat-alamat penyimpanan data kartu dan password khusus
pw0    data 20h
pw1    data 21h
pw2    data 22h
pw3    data 23h
pw4    data 24h
pw5    data 25h
pw6    data 26h
;pengisian data kartu dan password khusus
init:
        mov     pw0,#1ah;nomor kartu khusus 8730
        mov     pw1,#22h
        mov     pw2,#9  ;password khusus
        mov     pw3,#9
        mov     pw4,#6
        mov     pw5,#4
        mov     pw6,#5

```

Bank register dipilih dari bank nomor 3. Ini memberikan cukup ruang untuk digunakan sebagai stack. Bank ini akan ditempati oleh register-register 0 sampai 7. Register-register ini menyimpan data-data untuk pengalamanan tak langsung (indirect), loop pelambat.

```

;pilih bank register 3
  setb  rs0
  setb  rs1

```

Interrupt luar IE0 diprogram agar aktif pada mode *low level activated*. Bila catu daya utama mengalami kegagalan maka pin IE0 akan mendapat logika rendah dan mengaktifkan rutin pelayanannya.

Timer/counter dan saluran serial dipersiapkan untuk operasi komunikasi serial dengan data 8 bit dan kecepatan yang dapat diprogram. Untuk timer/counter 1 dioperasikan pada mode autoreload (mode 2) dan port serial pada mode 1.

```

mov  tmod,#00100000b ;counter 1 mode 2 (autoreload)
mov  t1l,#0 ;timer 1 lower byte
mov  th1,#-26;autoreload number
      ;serial port mode 3
      ;receive enable & unconditional
mov  scon,#11010000b
      ;run timer 1
      ;external interrupt low level activated

```

```

setb  trl      ;mov  tcon,#01000000b
                    ;enable serial port service routine
                    ;enable INT0 service routine
mov    IE,#10010001b
                    ;set int0 to a high interrupt priority
setb  ip.0

```

#### IV.3 Rutin pelayanan interrupt

Hanya ada dua interrupt yang dipakai dalam sistem ini. Yang pertama adalah interrupt prioritas tinggi, yakni interrupt untuk keadaan darurat karena kondisi catu daya. Rutin pelayanan interrupt ini terletak pada alamat 0003h. Rutin ini memberikan data "1" ke flag F0 (user flag) dan mematikan semua sumber interrupt. Dengan data "1" pada flag ini, maka data kartu dan password yang bisa diterima hanyalah data kartu dan password khusus. Bila keadaan catu daya normal kembali, maka rancangan perangkat keras akan membangkitkan sinyal reset dan program dimulai dari awal kembali.

Rutin pelayanan interrupt yang kedua mempunyai prioritas rendah. Rutin ini adalah rutin pelayanan untuk penanganan penerimaan data dari komunikasi serial. Sumber dari interrupt ini ada dua, yakni bit RI dan TI pada register SCON. Hanya permintaan interrupt yang bersumber dari RI sajalah yang dilayani.

Rutin yang terletak pada alamat 0023H ini mengerjakan pembukaan pintu ketika ada data yang diterima. Setelah pintu terbuka rutin menunggu sampai pintu tertutup kembali dengan melihat bit sensor P1.5. Dengan cara ini rutin tidak akan mengembalikan kontrol ke program yang diinterupsi selama pintu masih terbuka.

```

darurat bit psw.5 ;emergency condition bit
;rutin pelayanan keadaan darurat
    org    3
    clr    ea      ;clear interrupt flag
    setb   darurat
    reti
;rutin pelayanan penerimaan data komunikasi serial
    org    23h
    jnb    ri,terus;Ada data kriman ?
    clr    pintu   ;buka pintu
    acall  beep
    acall  delay
    acall  beep
    setb   pintu   ;matikan saklar
    jnb    pintus,$;tunggu sampai pintu ditutup
    clr    ri      ;clear interrupt flag
terus:reti

```

#### IV.4 Penanganan peralatan pendukung

Peralatan-peralatan pendukung terutama pembaca kode kartu dan pembaca tombol password ditangani dengan cara penelusuran (scanning). Cara ini menghemat pemakaian kaki-kaki port karena dapat dikonfigurasi sebagai matriks saklar. Matriks untuk pembaca kode kartu ditangani oleh kaki-kaki P1.0, P1.1, P1.2. Matriks untuk pembaca tombol password oleh kaki-kaki P1.3 dan P1.4.

Kaki-kaki tersebut berfungsi sebagai scan baris. Sedangkan untuk scan kolom digunakan perbandingan dengan perangkat lunak terhadap isi akumulator yang didapat dari port 0. Port 0 secara umum yang perangkat keras berfungsi sebagai BUS alamat dan data dihubungkan dengan keluaran dari gerbang three state buffer. Gerbang ini diaktifkan dengan sinyal pembacaan  $\overline{RD}$  pada saat pelaksanaan pembacaan data dari RAM luar. Karena sebenarnya RAM luar tidak ada, maka untuk pembacaan tersebut dapat dilakukan pada sembarang alamat RAM luar.

```

buzzer bit p1.7 ;buzzer bit
pintu bit p1.6 ;bit buka pintu
pintus bit p1.5 ;bit sensor pintu terbuka
kbd1 bit p1.4 ;keyboard scan bit 1
kbd0 bit p1.3 ;keyboard scan bit 0
card2 bit p1.2 ;bit 2 scan kartu
card1 bit p1.1 ;bit 1 scan kartu
card0 bit p1.0 ;bit 0 scan kartu
baca:
    mov r0,#0 ;alamat awal penyimpanan data
    clr card0 ;aktifkan sensor validitas kartu
    movx a,@r1 ;baca data dari peralatan sensor
    cjne a,#10100101b,baca ;periksa validitas kartu
    setb card0
    clr card1 ;baca nomor kartu byte pertama
    movx a,@r1 ;
    mov @r0,a ;simpan
    setb card1
    inc r0
    clr card2 ;baca nomor kartu byte kedua
    movx a,@r1
    mov @r0,a ;simpan
    inc r0
    setb card2
    acall beep
scan:
    clr card0 ;aktifkan sensor validitas kartu
    movx a,@r1 ;baca data dari peralatan sensor
    cjne a,#10100101b,baca ;periksa validitas kartu

```

```

    setb    card0    ;pastikan kartu masih ada
    mov     r4,#0    ;lihat tombol-tombol password
    clr     kbd0    ;baris pertama
    movx   a,@r1
    setb   kbd0
    cjne   a,#0,simpan;bila ada yang ditekan
    inc    r4       ;periksa dan simpan
    clr    kbd1    ;baris kedua
    movx   a,@r1
    setb   kbd1
    cjne   a,#0,simpan;bila ada yang ditekan
    sjmp   scan    ;periksa dan simpan

simpan:
    acall  beep
    cjne   r4,#0,second
    cjne   a,#1,k1
    mov    @r0,#0
    sjmp   simpans

k1:
    cjne   a,#2,k2 ;bila tombol 1 ditekan
    mov    @r0,#1
    sjmp   simpans

k2:
    cjne   a,#4,k3 ;bila tombol 2 ditekan
    mov    @r0,#2
    sjmp   simpans

k3:
    cjne   a,#8,k4 ;bila tombol 3 ditekan
    mov    @r0,#3
    sjmp   simpans

k4:
    cjne   a,#16,k5 ;bila tombol 4 ditekan
    mov    @r0,#4
    sjmp   simpans

k5:
    cjne   a,#32,k6 ;bila tombol 5 ditekan
    mov    @r0,#5
    sjmp   simpans

k6:
    cjne   a,#64,k7 ;bila tombol 6 ditekan
    mov    @r0,#6
    sjmp   simpans

k7:
    cjne   a,#128,scan ;bila tombol 7 ditekan
    mov    @r0,#7
    sjmp   simpans

second:
    ;tombol-tombol baris kedua ?
    cjne   a,#1,k9 ;bila tombol 8 ditekan
    mov    @r0,#8
    sjmp   simpans

k9:
    cjne   a,#2,kb ;bila tombol 9 ditekan
    mov    @r0,#9
    sjmp   simpans

```

```

kb:
    cjne a,#4,ke ;bila tombol R ditekan
    mov  r0,#0
    sjmp scan

simpans:
    inc  r0      ;teruskan sampai data terakhir
    cjne r0,#7,scan
    sjmp proses

ke:
    cjne a,#8,scan ;bila tombol E ditekan
    mov  @r0,#0
    inc  r0
    cjne r0,#7,ke

```

Tombol 0 - 9 adalah tombol nomor. Sedangkan tombol R (RESET) adalah tombol untuk pembatalan tombol-tombol password yang telah ditekan. Tombol E adalah tombol Enter. Tombol E digunakan untuk mengakhiri pemasukkan tombol password sekalipun belum lengkap lima tombol ditekan.

#### IV.5 Pengolahan data selanjutnya

Setelah data kode kartu dan password selesai dibaca, data-data tersebut diolah berdasarkan kondisi sistem saat itu. Dalam keadaan normal data-data langsung dikirimkan ke komputer basis data. Sedangkan dalam keadaan darurat data-data tersebut dibandingkan dengan data khusus yang ada di dalam RAM sendiri.

Dalam keadaan normal setelah data selesai dikirimkan, pekerjaan berikutnya adalah menunggu balasan dari

komputer basis data. Data balasan tersebut tidak dilihat isinya, tetapi hanya sinyal interrupt penerimaan data komunikasi serialnya saja yang dimanfaatkan (lihat rutin pelayanan interrupt serial komunikasi di atas).

```

proses:
    jnb    f0,lapor;bila keadaan darurat
    mov    r1,#26h ;bandingkan data dengan data khusus

cek:    dec    r0
        mov    a,@r0 ;jalan kedepan
        xrl    a,@r1
        jnz    cekend ;bila ada yang tidak cocok, maka
        dec    r1 ;bukan data masukan yang benar
        cjne   r1,#1fh,cek ;ulangi sampai data terdepan
        clr    pintu ;bila semua OK, buka pintu
        acall  delay
        jnb    pintus,$;tunggu sampai pintu ditutup
        setb   pintu

cekend:
        ajmp   baca ;baca data yang lain

lapor:
        setb   ti ;siapkan saluran kirim
        mov    r0,#0 ;mulai byte pertama

ulang:
        mov    a,@r0 ;muat di acc
        call   kirim ;kirirkan
        inc    r0 ;byte berikut
        cjne   r0,#7,ulang;sampai 7 byte
        jnb    ti,$ ;tunggu transmisi selesai
        clr    ti ;bersihkan saluran kirim
        ajmp   baca

kirim:
        mov    c,p ;letakkan parity di Carry
        mov    tb8,c ;lalu letakkan ke buffer kirim
        jnb    ti,$ ;tunggu saluran kirim siap
        clr    ti ;bersihkan saluran kirim
        mov    sbuf,a
        ret

```

#### IV.6 Beberapa rutin tambahan

Selain bagian utama program seperti tersebut di atas, masih ada beberapa rutin tambahan yang diperlukan untuk membantu pekerjaan bagian utama dari program

perangkat lunak ini. Rutin-rutin tersebut adalah rutin tunda (*Delay*) dan rutin suara (*Beeper*).

Rutin tunda terutama digunakan untuk memperpanjang waktu tanggap dari peralatan. Sebagai contoh adalah waktu tanggap dari peralatan pembuka pintu, peralatan pembaca tombol password dan lain-lain.

Rutin suara digunakan untuk memberi umpan balik kepada pemakai sebagai pemberi tanda bahwa peralatan telah menanggapi pemakai.

```

beep: clr    buzzer
      acall  delay
      setb   buzzer
      ret

delay: xrl   ie,#11h
      mov   r6,#0
      mov   r5,#80 ;set delay for w 100 ms
d1:    djnz  r6,$
      djnz  r5,d1
      xrl   ie,#11h
      ret

```

## BAB V

### SISTEM BASIS DATA

Basis data digunakan untuk mendapatkan informasi tentang kode kartu yang ada beserta password-nya. Data kartu dan password dari kontroller yang datang lewat komunikasi serial dicocokkan dengan data yang tersimpan dalam basis data. Bila ada dan cocok maka basis data mencatatnya dan memberikan perintah ke kontroller agar membuka pintu.

Basis data ditangani oleh sebuah komputer IBM-PC dan diprogram dengan bahasa pemrograman PASCAL. Di sini digunakan compiler Turbo PASCAL versi 6.0 dari Borland, dengan pertimbangan kemudahan dalam pengendalian port serial dan pengelolaan data.

Program penanganan basis data ditempatkan dalam file unit PERSON.PAS untuk memudahkan pengelolaan dan pengembangannya.

#### V.1 Bentuk data

Bentuk data dasar yang disimpan adalah sebuah record yang berisi tiga buah field. Field-field tersebut adalah field nomor kartu, kode password dan nama peme-

gang kartu. Bentuk record data tersebut adalah sebagai berikut :

```

const
  maxlist = 100; {asumsi jumlah data maksimum}
type
  stnm = string[30];
  st5 = string[5];
  data = record
    kartu : word;
    kode : st5;
    nama : stnm;
  end;
  DataPtr = ^data; {data dinamik dalam memori}
  dataArray = array[1..maxlist] of DataPtr; {array pointer}

```

## V.2 Penanganan data

Data dari setiap personil yang sudah dikenal disimpan dalam sebuah file dengan dikodekan, sehingga tidak dapat dibaca oleh program lain. Data-data ini diambil dari file pada saat inisialisasi program, dan disimpan setiap kali ada perubahan (tambah, ubah dan hapus).

Data personil yang baru dapat ditambahkan kapan saja. Data yang perlu dimasukkan hanyalah nama dan password-nya saja. Sedangkan nomor kartu secara otomatis diberikan oleh program dari pembangkit bilangan acak, dimana nomor tersebut sudah dipastikan tidak akan sama dengan nomor kartu lain yang sudah ada.

Di dalam memori, data ditempatkan di dalam record dinamik. Program dapat mengaksesnya lewat array pointer

yang menunjukkan lokasi-lokasi penyimpanannya. Hal ini mempertimbangkan bahwa pada proses pengurutan data, cara yang paling cepat adalah dengan memakai metoda *recursif quick sort* dengan data yang ditempatkan di dalam sebuah array pointer. Bila terjadi penukaran lokasi, maka yang ditukar hanyalah pointernya saja dan bukan isi datanya. Dengan demikian jumlah data yang dipindahkan dalam sekali pemindahan adalah seukuran data pointer saja, yaitu 2 word atau 4 byte.

Untuk kemudahan pemakaian program, disediakan dua buah array pointer. Satu array digunakan untuk pengurutan nomor kartu, dan yang lain untuk pengurutan nama. Pengurutan nama digunakan dalam pencarian data oleh operator sedangkan pengurutan nomor kartu digunakan untuk pencarian data dalam pengesahan data dari kontroler. Setiap terjadi perubahan data (tambah, ubah atau hapus) kedua array ini diurutkan kembali dan disimpan ke file.

Pencarian data dilakukan dengan metoda *binary tree search*. Metoda ini adalah metoda yang paling umum dan cepat untuk pencarian data yang tersimpan secara urut.

Baik pada pengurutan maupun pencarian data, kunci

data dapat ditentukan menurut nomor kartu maupun menurut nama. Hal ini dimungkinkan oleh kemampuan Turbo PASCAL dalam mengelola data jenis procedure. Untuk lebih jelasnya, lihat listing program berikut ini.

```

type lessfunc = function(x,y:dataptr): boolean;
var comp,less : lessfunc;
    fullname : string;

{$f+}
function compkartu(x,y : dataptr) : boolean;
begin
    compkartu := x^.kartu = y^.kartu;
end;

function compnama(x,y : dataptr) : boolean;
begin
    compnama := x^.nama = y^.nama;
end;
function lesskartu(x,y : dataptr) : boolean;
begin
    lesskartu := x^.kartu < y^.kartu;
end;

function lessnama(x,y : dataptr) : boolean;
begin
    if compnama(x,y) then
        lessnama := (x^.kartu < y^.kartu)
    else
        lessnama := (x^.nama < y^.nama);
end;{$f-}

procedure setfunction(nomor : byte);
{procedure untuk menentukan kunci pengurutan atau pencarian}

begin
    metode := nomor;
    case nomor of
    1 : begin
        @less := @lesskartu;
        @comp := @compkartu
    end;
    2 : begin
        @less := @lessnama;
        @comp := @compnama
    end;
end;

```

```

        end;
    else
        begin
            @less := nil;
            @comp := nil
        end
    end;
end;
end;

```

```

procedure sort(var a: dataarray);
{procedure pengurutan dengan metode recursiv quick sort}
procedure quick(l,r: integer);
var
    x    : dataptr;
    i,j  : integer;
    y    : pointer;
begin
    i:=l; j:=r; x:=a[(l+r) DIV 2];
    repeat
        while less(a[i],x) do inc(i);
        while less(x,a[j]) do dec(j);
        if (i<=j) then
            begin
                y:=a[i]; a[i]:=a[j]; a[j]:=y;
                INC(I);DEC(J);
            end
        until i>j;
        if l<j then quick(l,j);
        if i<r then quick(i,r)
    end;
begin {sort};
    if jumlah > 1 then
        quick(1,jumlah)
    end;
end;

```

```

function search(list : dataarray;dt : data) : integer;
{rutin pencarian data dengan metode binary tree search}
var atas,tengah,bawah : integer;
    ketemu              : boolean;
    dlocal              : dataptr;

begin
    bawah := 1;
    atas  := jumlah;
    ketemu := false;
    dlocal := @dt;
    while not ketemu and (bawah <= atas) do
        begin

```

```
tengah := (atas+bawah) div 2;
ketemu := comp(list[tengah],dlocal);
if not ketemu then
  if less(dlocal,list[tengah]) then
    atas := tengah - 1
  else
    bawah := tengah + 1
end;
if ketemu then
  search := tengah
else
  if less(list[tengah],dlocal) then
    search := -tengah-1
  else
    search := -tengah
end;
```

### V.3 Kerahasiaan data

Semua data yang ada dapat dimunculkan kembali hanya dengan diketahuinya password bagi data yang bersangkutan. Tanpa password, data tidak dapat diakses untuk diubah ataupun dihapus. Untuk penambahan data baru password harus diketikkan dua kali. Bila ada ketidakcocokan diantara keduanya, maka password dianggap batal. Data password hanya bisa dilihat dan diubah pada saat pengubahan data. Dengan cara ini sebuah data hanya bisa diubah atau dihapus dengan sepengetahuan personil yang memilikinya. Untuk penanganan data yang mendesak, disediakan sebuah password khusus yang berlaku untuk semua data. Password ini juga diperlukan untuk penghapusan data secara menyeluruh. Password ini tidak disimpan di dalam file data, tetapi di dalam file program ini sendiri.

Penyimpanan data di dalam file melalui tahap pengkodean. Metode yang digunakan dalam pengkodean di sini sederhana saja, yaitu dengan pemutaran (*rotate*) bit-bit dalam setiap byte data. Pada saat penyimpanan, dilakukan pemutaran tiga bit ke kiri. Sebaliknya pada saat pembacaan, bit-bit diputar ke kanan.

```

procedure rebuildistnama;
  {procedure ini digunakan untuk mengurutkan kembali data dalam
   array pointer dengan kunci nama}
begin
  listn := list;
  setfunction(nama);
  sort(listn)
end;

function password(kode : st5) : boolean;
  {Rutin untuk mengesahkan akses ke data yang dimaksud}
  var i,j : word;
      ch : char;
      stc,std : string[5];

begin
  textbox(30,11,51,15,79, '');
  center(12, 'Masukkan Kode', 78);
  stc := kode;
  center(14, '00000', 78);
  movecursor(38,14);
  setcursor(off);
  textattr := 112;
  ch := #32;
  i := 0;
  while (ch <> #13) and (i < 5) do
  begin
    ch := readkey;
    if ch <> #13 then
      begin

```

```

        write('0');
        inc(i);
        std[i] := upcase(ch)
    end
end;
for j := i+1 to 5 do write('0');
std[0] := char(i);
{disahkan hanya bila pasword sesuai dengan data atau password
khusus yaitu yang tersimpan dalam variabel pass}
password := (std = stc) or (std = pass);
closebox
end;

```

```

procedure newlist;
{procedure untuk menghapus file data. Semua data yang lama
akan dihapus }
var batal : boolean;
    f : file;
begin
    if jumlah > 0 then
        begin
            textbox(20,5,60,11,128+78,'Peringatan');
            setcursor(off);
            write(#7#7);
            textattr := 78;
            center(7,'Semua data di file akan dihapus',78);
            center(9,'Lanjutkan ? [Y]/[T]',78);
            batal := upcase(readkey) <> 'Y';
            closebox;
            if batal or not password(pass) then exit;
                {Bila password tidak sesuai maka dibatalkan}
                {Bila sesuai maka semua data dinamik dibuang}
            while jumlah > 0 do
                begin
                    dispose(list[jumlah]);
                    dec(jumlah)
                end;
                {Hapus file data}
            assign(f,fnama);
            setfattr(f,$20);
            erase(f);
        end
    end;
end;

```

```

procedure simpanlist;
{Procedure untuk menyimpan data}
var i,j,L : integer;

```

```

    f.: file of data;
    g : data;
type sandi = array[1..sizeof(data)] of byte;
  {digunakan untuk pengkodean data}
begin
  fullname := fsearch(fnama, '');
  if fullname = '' then
    assign(f, fnama)
  else
    begin
      assign(f, fnama);
      setfattr(f, $20)
    end;
  rewrite(f);

  for I := 1 to jumlah do
    begin
      g := list[i]^;
      {pengkodean dilakukan di sini}
      for j := 1 to sizeof(data) do
        begin
          {diutar ke kiri tiga bit}
          l := sandi(g)[j] shr 5;
          sandi(g)[j] := sandi(g)[j] shl 3 + l
        end;
      write(f, g)
    end;
  close(f);
  listchanged := false;
  setfattr(f, attr)
end;

procedure loadlist;
{procedure mengambil data dari file}
var f : file of data;
    i, L : byte;

type sandi = array[1..sizeof(data)] of byte;
begin
  fullname := fsearch(fnama, '');
  listOk := 0;
  if fullname = '' then
    begin
      ListOk := 2; {file not found}
      exit
    end;
  assign(f, fnama);
  setfattr(f, $20);
  reset(f);

```

```

{buang semua data di memori}
while (jumlah > 0) do
begin
  dispose(list[jumlah]);
  dec(jumlah)
end;
{ambil data di file}
maxlen := 0;
while not eof(f) do
begin
  inc(jumlah);
  new(list[jumlah]);
  read(f,list[jumlah]^);
  {data diterjemahkan di sini}
  for i := 1 to sizeof(data) do
  begin
    {putar 3 bit ke kanan}
    l := sandi(list[jumlah]^)[i] shl 5;
    sandi(list[jumlah]^)[i] := sandi(list[jumlah]^)[i] shr 3 + l
  end;
  if maxlen < length(list[jumlah]^).nama) then
    maxlen := length(list[jumlah]^).nama
end;
setfattr(f,attr);
listchanged := false;
rebuildlistnama
end;

```

#### V.4 Penambahan data

Penambahan data baru dilakukan dengan cara menyisipkan data pada tempat yang benar dalam urutan. Ini dilakukan dengan cara membandingkan nomor kartunya dengan nomor kartu yang lain. Nomor kartu seperti yang telah dibicarakan di atas, dihasilkan oleh suatu fungsi dengan sebuah pembangkit bilangan acak. Bilangan tersebut dibandingkan dengan nomor-nomor kartu yang sudah ada untuk memastikan bahwa nomor kartu yang sama tidak akan ada. Pencarian tempat dalam urutan dilakukan dengan

seolah-olah dilakukan pencarian data. Rutin pencarian data akan membawa penunjuk record ke tempat terdekat yang lebih besar dari data yang akan ditambahkan. Kemudian data baru ditempatkan di sini, dan data-data lain yang lebih besar darinya digeser ke belakang.

```

function insertlist(dt : data):byte;
{Rutin ini digunakan untuk menyisipkan data baru
 pada tempat yang tepat dalam urutan}
var dlocal : dataptr;
    cek      : boolean;
    nm,i     : integer;

begin
    cek := true;
    i := 1;
    if jumlah >= 1 then
        begin
        {cari tempat yang tepat dalam urutan}
            setfunction(kartu);
            i := search(list,dt);
        {periksa apakah ada data yang sama}
            cek := i < 1
        end;
        {bila ternyata ada yang sama maka harus ditolak}
        if not cek or (dt.kartu = 0) then
            begin
                insertlist := 20; { nomor kartu tidak unik }
                exit
            end;
        {persiapkan tempat data dinamik di memori}
            new(dlocal);
            if dlocal = nil then
                begin
                {bila gagal maka kemungkinan memori telah penuh}
                    insertlist := 2; { memori tidak cukup }
                    exit
                end;
                inc(jumlah);
                if maxlen < length(dt.nama) then maxlen := length(dt.nama);
                nm := jumlah;
                if i < 0 then i := -i;

```

```

dlocal^.kartu := dt.kartu;
dlocal^.kode := dt.kode;
dlocal^.nama := dt.nama;
{tempatkan data baru di akhir urutan}
list[nm] := dlocal;
{kemudian digeser ke depan sampai pada urutan yang benar}
while (nm > i) do
begin
list[nm] := list[nm-1];
dec(nm);
list[nm] := dlocal
end;
listchanged := true;
insertlist := 0
end;

function CariNomor : word;
{Rutin untuk mencari nomor kartu yang unik dengan
pembangkit bilangan acak}
var i,j : word;
begin
randomize;
inline($fa);
repeat
j := random(65535);{pembangkitan bilangan acak}
i := 1;
{pastikan bilangan tidak sama dengan nomor kartu yang lain}
while (i <= jumlah) and (j <> list[i]^kartu) do inc(i)
{ulangi sampai benar}
until i > jumlah;
inline($fb);
carinomor := j
end;

procedure AddList;
{procedure untuk menambahkan data baru}
var sk : string[5];
j : integer;
i : word;
d : data;

begin
textbox(4,4,45,10,30+dobel, 'Tambah data');
sets(delimiter,off);
TEXTATTR := 31;
gotoxy(11,2);
sets(warna,on);
sets(kapital,on);
setcursor(on);

```

```

sets(jenis,huruf);
{cari nomor kartu yang unik dan tampilkan}
d.kartu := carinomor;
write(' Nomor kartu : ',d.kartu:4);
{ambil data nama dan password}
write('^j^J^m' Nama : ');
d.nama := instring('',30);
if d.nama <> '' then
begin
sets(jenis,angka);
sets(secret,on);
sets(warna,on);
textbox(30,8,51,16,79,'Kode Password');
repeat
clrscr;
gotoxy(8,2);
{ambil data password}
sk := instring('',5);
center(12,'Ketikan kembali',78);
{ulangi untuk pengesahan}
gotoxy(8,6);
{ulangi memasukkan data password sampai keduanya cocok }
until (sk=instring('',5));
sets(secret,off);
closebox;
d.kode := sk;
end;
closebox;
if d.nama <> '' then
begin
{sisipkan ke urutan}
j := InsertList(d);
rebuilddlistnama
end
end;

```

## V.5 Mengubah dan menghapus data

Untuk mengubah dan menghapus data yang sudah ada, maka data tersebut harus diketahui identitasnya. Yang paling mudah untuk diingat dalam pencarian ini adalah data nama. Oleh karena itu disediakan juga rutin yang dapat menampilkan dan membantu pemilihan data nama

tersebut. Rutin ini menampilkan data secara dinamik, dalam arti bila jumlah semua data kurang dari 12 maka jumlah baris scroll bar yang ditampilkan hanya sejumlah datanya saja. Bila lebih, maka scroll bar dibatasi pada 12 baris, sedangkan data-data selebihnya dapat ditampilkannya dengan penggulungan layar.

```

function pilihlist(nm : stnm) : word;
{rutin pembantu pemilihan data dengan menu scroll bar}
var bawah,atas,maxbaris,bbw,bba,pilih,i : integer;
    ch,cr : char;
    done, redraw : boolean;
    dt : data;
    st,sj : string[40];

begin
    bawah := 0;
    if nm <> '' then
        begin
            dt.nama := nm;
            setfunction(nama);
            bawah := search(listn,dt);
            if bawah > 0 then
                begin
                    atas := bawah;
                    while (bawah > 1) and (listn[bawah-1]^nama = nm) do dec(bawah);
                    while (atas < jumlah) and (listn[atas+1]^nama = nm) do inc(atas)
                end
            end;
            pilih := bawah;
            if bawah < 1 then
                begin
                    bawah := 1;
                    atas := jumlah
                end;
            {jumlah baris scroll bar maksimal adalah 12 baris}
            maxbaris := 12;
            {bila jumlah data kurang dari 12, maka jumlah baris scroll bar
            sama dengan jumlah data}
            if atas - bawah < maxbaris then
                maxbaris := atas - bawah + 1;
            textbox(30,4,39+MAXLEN,5+maxbaris,48,'Daftar ');
        end
    end;
end;

```

```

setcursor(off);
bbw := bawah ;
done := false;
if (pilih < 0) and (-pilih <= jumlah) then
begin
  pilih := -pilih;
  while pilih > maxbaris do
  begin
    inc(bbw);
    dec(pilih)
  end
end
else
  pilih := 1;
textattr := 31;
Sj := DUPCHR(' ',8+MAXLEN);
repeat
  bba := bbw + maxbaris - 1;
  for i := bbw to bba do
    with listn[i]^ do
      BEGIN
        STR(kartu:6,nm);
        st := sj;
        st := nm + ' ' + nama + ' ';
        ST[0] := sj[0];
        prinstr(31,5+i-bbw,ST,textattr)
      end;
  repeat
    redraw := false;
    setattr(79,31,5+pilih-1,38+MAXLEN,5+pilih-1);
    ch := readkey;
    case ch of
      #0 : begin
        cr := readkey;
        if cr in [#72,#80] then
          setattr(textattr,31,5+pilih-1,38+MAXLEN,5+pilih-1);
        case cr of
          {gerakkan scroll bar ke atas}
          #72 : if pilih > 1 then dec(pilih)
                else
                  if bbw > bawah then
                    begin
                      dec(bbw);
                      redraw := true
                    end
                  else
                    begin
                      bbw := atas - maxbaris + 1;
                      pilih := maxbaris;

```

```

        redraw := true
    end;
    {gerakkan scroll bar ke bawah}
    #80 : if pilih < maxbaris then inc(pilih)
        else
            if bba < atas then
                begin
                    inc(bbw);
                    redraw := true
                end
            else
                begin
                    bbw := bawah;
                    pilih := 1;
                    redraw := true
                end
            end
        end
    end;
    #13,#27: done := true
    end
    until done or redraw
until done;
closebox;
if ch = #13 then
    pilihlist := bbw + pilih - 1
else
    pilihlist := 0
end;
end;

function CariRecord : word;
{rutin pencarian data}
var i : integer;
    nm : stnm;
    dt : data;

begin
    if jumlah = 0 then
        begin
            ListOk := 21; { list empty }
            Carirecord := 0;
            exit
        end;
    textbox(25,8,58,13,31+dobel,'Cari record');
    clrscr;
    {ambil data nama}
    writeln('^J^M' Nama : ');
    setattr(15,27,11,56,11);
    textattr := 15;
    gotoxy(2,3);
    sets(warna,off);

```

```

sets(jenis,huruf);
sets(delimiter,off);
sets(kapital,on);
setcursor(on);
NM := instring('',30);
i := pilihlist(nm);
if i > 0 then
begin
dt.kartu := listn[i]^kartu;
setfunction(kartu);
carirecord := search(list,dt)
end
else
carirecord := 0;
closebox
end;

```

Pengubahan data dilakukan hanya dengan mengganti isi dari variabel penyimpan data. Sebelum dapat menggantinya, operator diharuskan dapat memasukkan password data yang bersangkutan terlebih dahulu. Bila tidak dapat, maka penggantian dibatalkan.

Data yang diubah hanyalah data nama dan password saja. Untuk ini data password diperlihatkan untuk memudahkan perubahan.

Setelah perubahan dilakukan, maka urutan data dengan nama sebagai kunci harus diperbaiki kembali.

```

Procedure EditList(var dt : data);
{rutin untuk mengubah isi data}
var sk : string[5];

```

```

    sn : stnm;
    j,nm,l : integer;
    i : word;
    s : boolean;

begin
    setfunction(kartu);
    nm := search(list,dt);
    if (nm < 1) or not password(dt.kode) then
    begin
        write(#7);
        exit
    end;
    textbox(4,4,45,10,30+dobel, ' Edit data ');
    sets(delimiter,off);
    setcursor(on);
    write('^j' Nomor kartu : ',dt.kartu:4);
    sk := dt.kode;
    writeln(' Nomor kode : ');
    write('^j' Nama : ',dt.nama);
    TEXTATTR := 31;
    sets(warna,on);
    sn := dt.nama;
    repeat
        sets(jenis,angka);
        gotoxy(35,2);
        sk := instring(sk,5);
        sets(jenis,huruf);
        gotoxy(9,4);
        sn := instring(sn,30);
    until selesai;
    dt.kode := sk;
    dt.nama := sn;
    if length(sn) > maxlen then maxlen := length(sn);
    closebox;
    listchanged := true;
    REBUILDLISTNAMA
end;

```

Hal yang serupa juga dilakukan untuk menghapus data. Setelah operator dapat memasukkan data password dengan benar, barulah data yang bersangkutan dihapus.

Penghapusan data dilakukan dengan mengganti isi data yang akan dihapus dengan isi data di urutan ter-

akhir. Kemudian tempat data diurutan terakhir inilah yang dihilangkan dari memori. Setelah itu urutan pointer disusun kembali. Dengan cara ini dapat dicegah ruang memori yang berlubang akibat penghapusan, karena tempat penyimpanan data yang dibebaskan selalu yang terakhir.

```

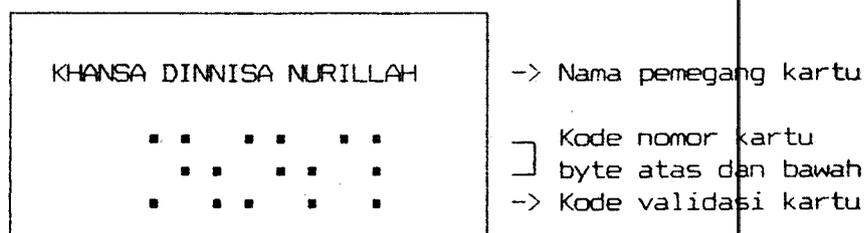
procedure hapuslist(dt : data);
  {rutin untuk menghapus data}
  var dlocal : dataptr;
      nm,i    : integer;
  begin
    if dt.kartu = 0 then exit;
    dlocal := @dt;
    setfunction(kartu);
    nm := search(list,dt);
    if (nm < 1) or not password(list[nm]^kode) then
      begin
        write(#7);
        exit
      end;
    {pindahkan isi data di urutan terakhir ke tempat data yang
    akan dihapus}
    dlocal := list[nm];
    dlocal^ := list[jumlah]^;
    {susun kembali urutan pointer}
    while nm < jumlah do
      begin
        inc(nm);
        list[nm-1] := list[nm]
      end;
    list[nm] := dlocal;
    {bebaskan tempat penyimpanan data terakhir}
    dispose(list[jumlah]);
    dec(jumlah);
    maxlen := 0;
    for i := 1 to jumlah do
      if maxlen < length(list[i]^nama) then
        maxlen := length(list[i]^nama);
    listchanged := true;
    rebuildlistnama
  end;

```

## V.6 Pencetakan bentuk kartu

Pencetakan ini ditujukan untuk memberikan pola dari bentuk kartu yang akan dibawa oleh personil. Pola ini menyangkut bentuk dan kode optik yang harus dibuat untuk masing-masing kartu.

Pada masing-masing kartu hanya terdapat nama pemilik kartu dan kode optik saja. Kode optik terdiri dari tiga baris. Satu baris untuk kode validasi kartu dan dua yang lain adalah nomor kartu. Kode validasi selalu sama untuk setiap kartu.



Gambar 5.1  
Bentuk pola kartu

```

procedure cetakkartu(dt : data);
const n = #27'4'#27'E'#27'G';
      f = #27'5'#27'F'#27'H';
      l = #27'0'#27'U1';
      k = #27'1';
      d : array[0..1] of char = ' [';

var   lb,lb1,lb2: stnm;
      nomor : word;
      g,h,i,j : byte;
      nm : integer;
      ch : char;
      mwc:pwindscr;

```



## V.7 Penanganan komunikasi serial

Komunikasi serial ini diperlukan untuk menerima data dari kontroller untuk diperiksa dan disahkan. Selain itu juga untuk mengirimkan data pengesahan yang memberikan perintah untuk membuka pintu. Jadi hanya bila data nomor kartu sesuai dengan password data pengesahan dikirimkan ke kontroller.

Komunikasi serial diselenggarakan dengan bantuan pelayanan interrupt yang disediakan oleh BIOS. Detil dari pelayanan-pelayanan yang ada sudah dibicarakan pada bab III, tentang port serial pada komputer IBM-PC.

Rutin-rutin yang digunakan dalam penanganan komunikasi serial ini semuanya tercantum dalam file unit AU.PAS.

```

unit au;
interface
const baud300 = $4A; {Semua parameter diset dalam parity}
      baud600 = $6A; {ganjil dengan ukuran karakter 7 bit}
      baud1200 = $8A; {dan satu stop bit}
      baud2400 = $AA;
      baud4800 = $CA;
      baud9600 = $EA;
type st10 = string[5];
      comdata = record
          kartu : word;
          kode : st10
      end;
const nomor : word = 1;
      baud : word = baud1200;
      comavail : boolean = false;
var combuffer : comdata;

```

```
procedure initcom(port,params : word);
function comready : boolean;
function incom : char;
procedure outcom( ch : char );
procedure readcom (var data : comdata);
procedure writecom(s : sti0);
procedure checkcom;
implementation
uses crt,dos;
```

```
procedure initcom(port,params : word);
begin
  cominit := true;
  nomor := port;
  baud := params;
  asm
    mov ax,params
    mov dx,port
    mov ah,0
    int $14
  end
end;
```

```
function comready:boolean;
var i : byte;
    port : word;
begin
  port := nomor;
  asm
    mov dx,port
    mov ah,3
    int $14
    mov i,ah
  end;
  comready := i and 1 = 1
end;
```

```
function incom : char;
var j : char;
    port : word;
begin
  port := nomor;
  asm
    mov dx,port
    mov ah,2
    int $14
    mov j,al
  end;
  incom := j
end;
```

```

procedure outcom(ch:char);
var port : word;
begin
  port := nomor;
  asm
    mov al,ch
    mov dx,port
    mov ah,1
    int $14
  end
end;

procedure readcom (var data : comdata);
var j : comdata;
    k : array[1..8] of char absolute j;
    h,i : byte;
begin
  data.kartu := 0;
  data.kode := '';
  if not comavail then exit;
  inline($fa);
  i := 0;
  repeat
    inc(i);
    while not comready do;
      k[i] := incom;
  until (k[i] = #13) or (i = 8);
  inline($fb);
  if k[i] <> #13 then exit;
  for h := i downto 4 do
    k[h] := k[h-1];
  k[3] := chr(i-3);
  data := j
end;

procedure writecom(s : st10);
var i : byte;
begin
  for i := 1 to length(s) do
    outcom(s[i])
end;

procedure checkcom;
begin
  comavail:= comready;
  if comavail then readcom(combuffer);
end;
begin
  initcom(1,baud1200);
end.

```

### V.8 Program utama

Program sistem basis data ini dapat dibagi menjadi dua bagian besar. Yaitu bagian penanganan pengesahan kode dan password dari kontroller beserta pencatatan dan pengiriman kode pengesahan ke kontroller, dan bagian penanganan basis data dari personil-personil yang mempunyai kewenangan untuk memasuki ruangan.

Program utama dari sistem basis data hanya berisi pengaturan yang diperlukan untuk melakukan kedua hal di atas.

Unit-unit yang tercantum dalam listing program utama ini selain unit PERSON dan AU merupakan unit pembantu tampilan.

```
uses crt,dos,tscreen,person,au,menu;

var f : text;
    namafile : string[12];
    menu1 : menupopptr;
Const m1 : byte = 25;

procedure catat(dt : data; open : boolean);
{rutin pencatat laporan dari kontroller}
var hour,menit,detik,dtk100 : word;
    name : strn;

const status : array[false..true] of string[5]
    = ('Gagal','Buka');
```

```

begin
  if dt.kartu = 0 then exit;
  fillchar(name,31,32);
  move(dt.nama,name,length(dt.nama));
  name[0] := #30;
  gettime(hour,menit,detik,dtk100);
  if open then outcom(#1);
  writeln(dt.kartu:5,' ',name,' ',hour:2,':',menit:2,':',
          detik:2,' ',status[open]);
  writeln(f,dt.kartu:5,' ',name,' ',hour:2,':',menit:2,':',
          detik:2,' ',status[open]);
  flush(f);
end;

procedure database;
{rutin penanganan basis data personal}
var ch : integer;
    i : word;

begin
  repeat
    savescr(25,ml,55,ml);
    center(ml,#24#25'Pindah bar '#17#196#217'Pilih item',95);
    if jumlah = 0 then menu1^.disable([1,3,4,5])
    else menu1^.able([1,3,4,5]);
    ch := menu1^.popup;
    restscr;
    case ch of
      1 : newlist;
      2 : addlist;
      3..5 : begin
        i := carecord;
        if i > 0 then
          case ch of
            3 : editlist(list[i]^);
            4 : hapuslist(list[i]^);
            5 : cetakkartu(list[i]^);
          end
        end
      end
    end
  until ch = 7;
  if listchanged then simpanlist;
end;

procedure kerja;
{rutin yang merupakan inti dari program utama}
var d,m,y,dow: word;
    sd,sm,sy : string[4];
    i : integer;
    dt : data;
    ch,cr : char;

```

```

const selesai : boolean = false;
    hari : array[0..6] of string[6] =
        ('Minggu', 'Senin', 'Selasa', 'Rabu', 'Kamis', 'Jumat',
         'Sabtu');

function yakin : boolean;
{rutin untuk meyakinkan operator menginginkan
mengakhiri program}

begin
textbox(31,12,49,14,48, '');
write(' Selesai ? (Y/T)');
yakin := upcase(readkey) = 'Y';
closebox
end;

begin
if stype = 1 then
begin
line26(on);
ml := 26
end;
border(7);
textattr:=7;
cls(#219);
setattr(3,1,5,80,ml);
textattr:=30;
window(2,6,79,ml-1);
clrscr;
center(1, ' SISTEM DATABASE KUNCI ELEKTRONIK ',95);
printr(1,ml,dupchr(#32,80),63);
printr(1,3,dupchr(#205,80),63);
CENTER(ml, '[B]asis data [S]eleesai ',56);
initcom(1,baud1200);
getdate(y,m,d,dow);
str(d,sd);
str(m,sm);
str(y,sy);
center(3, ' Catatan pintu, hari '+hari[dow]+' '+sd+'-'+sm+'
'+sy+' ',120);
namafile := sd+sm+sy+'.pnt'; {namafile sesuai tanggal}
assign(f,namafile);

if fsearch(namafile, '') = '' then
rewrite(f)
else
append(f);
{persiapkan menu penanganan basis data}
new(menu1,
init(30,6,91+dobel,95,95,1,94,88, ' Basis Data '));

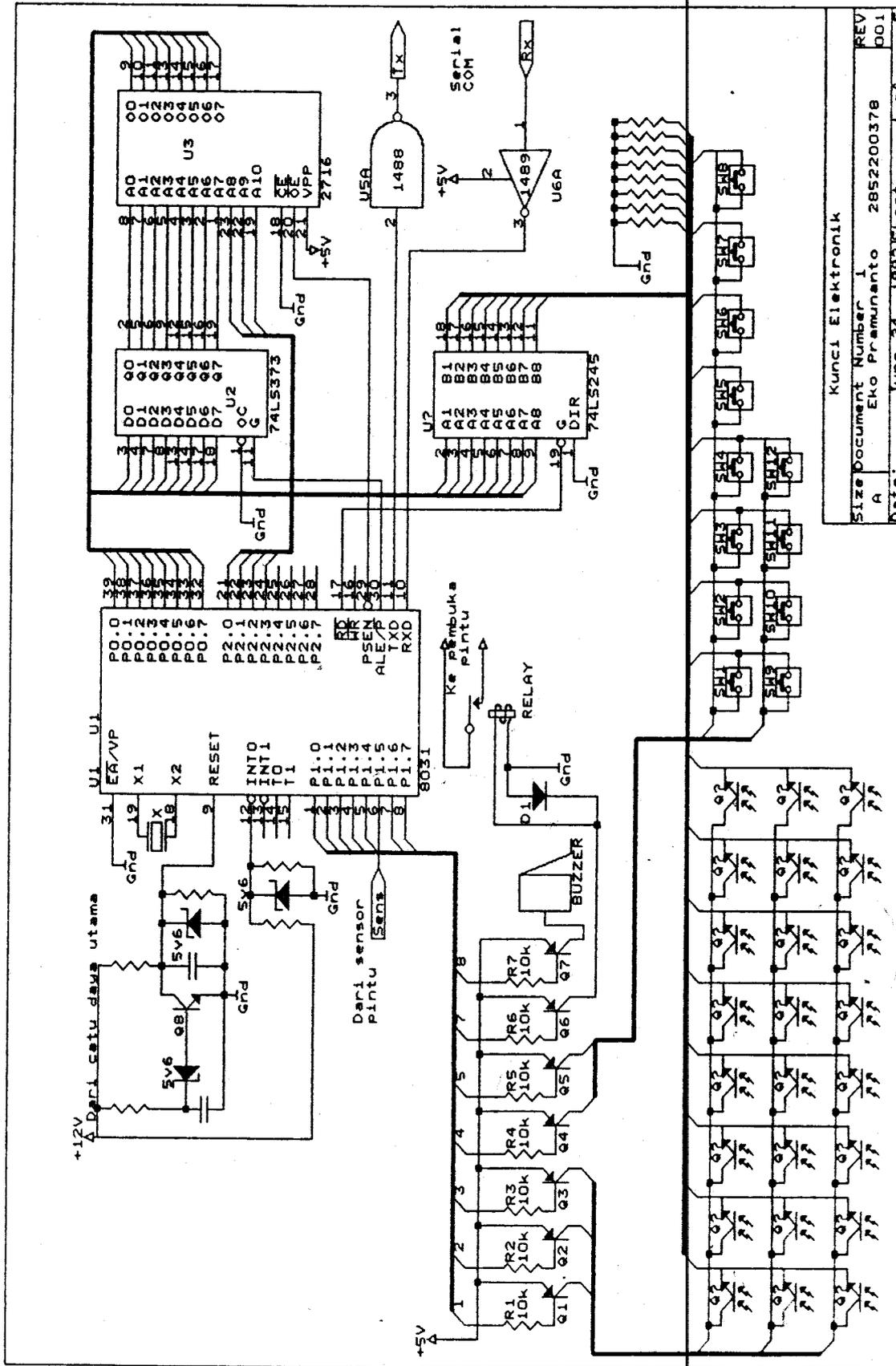
```

```

with menu1^ do
begin
  newitem;
  additem('File data ^baru');
  additem('^Tambah data');
  additem('^Ubah data');
  additem('^Hapus data');
  additem('^Cetak data');
  addline;
  additem('^Selesai');
end;
repeat
  if jumlah > 0 then
  begin
    {periksa laporan dari kontroller}
    Checkcom;
    {bila ada periksa dan catat}
    if comavail then
    begin
      comavail := false;
      dt.kartu := combuffer.kartu;
      dt.kode := combuffer.kode;
      setfunction(kartu);
      i := search(list,dt);
      {bila data yang dimaksud ada, maka catat}
      if i > 0 then
        catat(list[i]^,checkkode(dt.kartu,dt.kode))
      end
    end;
    {kemungkinan operator menginginkan sesuatu}
    if keypressed then
    begin
      ch := upcase(readkey);
      if ch = 'B' then database;
      selesai := (ch = 'S') and yakin
    end
  until selesai;
  dispose(menu1,done);
  close(f)
end;

{program utama}
begin
  kerja;
end.

```



|                      |                          |
|----------------------|--------------------------|
| Kunci Elektronik     |                          |
| Size Document Number | 1                        |
| A                    | Eko Prumnanto 2852200378 |
| Date:                | June 24, 1992            |
| Sheet                | 1 of 5                   |
| REV                  | 001                      |

FAKULTAS TEKNOLOGI INDUSTRI  
JURUSAN TEKNIK ELEKTRO ITS

EE. 1799 TUGAS AKHIR - 6 SKS

No. Pokok dan Nama Mahasiswa : 2852200378 / EKO PRAMUNANTO  
Bidang Studi : Teknik Komputer  
Tugas diberikan : 20. Maret. 1991.....  
Tugas diselesaikan : .....  
Dosen pembimbing : Dr.Ir. Handayani Tj., MSc

JUDUL TUGAS AKHIR :

SISTEM KUNCI ELEKTRONIK BERBASIS DATA  
DENGAN MENGGUNAKAN MIKROKONTROLER 8031

URAIAN TUGAS AKHIR :

Ruangan yang bersifat privat semakin dibutuhkan keberadaannya, baik untuk keperluan perorangan maupun kelompok. Ruangannya semacam ini terutama dibutuhkan sebagai tempat penyimpanan ataupun sebagai pusat kontrol dimana tidak sembarang orang boleh memasukinya. Untuk itu ruang tersebut haruslah diperlengkapi dengan sistem pengaman yang berupa kunci yang dapat berupa kunci mekanik konvensional ataupun kunci elektronik yang canggih.

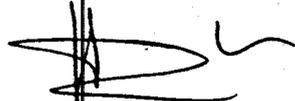
Pada tugas akhir ini akan dikemukakan kunci elektronik dengan minimum sistem 8031 yang diperlengkapi dengan sensor optik dan tombol password serta dihubungkan pada sistem basis data. Dengan demikian orang yang berada dalam ruangan tersebut dapat dicatat kapan masuk dan kapan keluarnya.

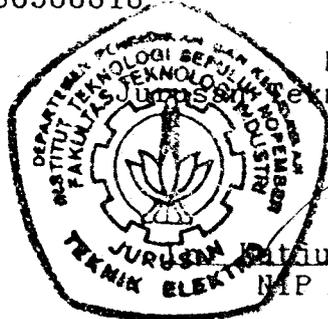
Surabaya, 1 Februari 1991

Menyetujui,  
Bidang studi komputer  
koordinator,

Dosen pembimbing

  
Dr. Ir. Soepeno Dianali, MSc  
NIP : 130368610

  
Dr. Ir. Handayani Tj., MSc  
NIP : 130532048



Mengetahui  
Teknik Elektro FTI - ITS  
Ketua,

  
Ketua Astrowulan, MSEE  
NIP : 130 687 438

## USULAN TUGAS AKHIR

1. JUDUL : SISTEM KUNCI ELEKTRONIK BERBASIS DATA DENGAN MENGGUNAKAN MIKRO-KONTROLER 8031
2. BIDANG STUDI : TEKNIK KOMPUTER
3. RUANG LINGKUP :
  - TEKNIK DIGITAL
  - MIKROPROSESOR
  - BAHASA ASEMBLY
  - BASIS DATA
4. LATAR BELAKANG : Sistem kunci elektronik akan memberikan sistem pengamanan yang cukup handal. Rancangan sistem yang sederhana secara hardware akan sangat memudahkan pembuatan serta perawatannya, karena sistem ini menggunakan mikrokontroler 8031.
5. PENELAAHAN STUDI : Prinsip kerja sistem ini sbb. :
  - Sensor optik  
Sensor ini digunakan untuk membaca kartu yang mempunyai lubang-lubang optik dan diatur sedemikian rupa sehingga membentuk suatu kode optik yang spesifik.
  - Tombol password  
Tombol password digunakan untuk memasukkan kode nomor. Kode ini ke-

## USULAN TUGAS AKHIR

1. JUDUL : SISTEM KUNCI ELEKTRONIK BERBASIS DATA DENGAN MENGGUNAKAN MIKRO-KONTROLER 8031
2. BIDANG STUDI : TEKNIK KOMPUTER
3. RUANG LINGKUP :
  - TEKNIK DIGITAL
  - MIKROPROSESOR
  - BAHASA ASEMBLY
  - BASIS DATA
4. LATAR BELAKANG : Sistem kunci elektronik akan memberikan sistem pengamanan yang cukup handal. Rancangan sistem yang sederhana secara hardware akan sangat memudahkan pembuatan serta perawatannya, karena sistem ini menggunakan mikrokontroler 8031.
5. PENELAAHAN STUDI : Prinsip kerja sistem ini sbb. :
  - Sensor optik  
Sensor ini digunakan untuk membaca kartu yang mempunyai lubang-lubang optik dan diatur sedemikian rupa sehingga membentuk suatu kode optik yang spesifik.
  - Tombol password  
Tombol password digunakan untuk memasukkan kode nomor. Kode ini ke-

mudian dicocokkan dengan kode nomor yang diambil dari sistem basis data di sebuah komputer IBM-PC yang dihubungkan dengan sistem ini.

- Basis data

Sistem kunci dihubungkan dengan sebuah komputer IBM-PC yang menangani basis data. PC menerima data kode kartu dari sistem kunci melalui port serial. Kode ini kemudian diperiksa keberadaannya, dan setelah ternyata ada maka PC akan menjawab dengan kode nomor password yang harus dimasukkan melalui tombol password di sistem kunci.

6. TUJUAN

: Kunci elektronik dengan sistem yang dilengkapi dengan basis data memungkinkan terselenggaranya sistem pengamanan ruangan yang lebih handal.

7. RELEVANSI

: Sistem ini mungkin dikembangkan untuk menangani pengamanan ruangan yang mempunyai beberapa pintu, atau bisa juga beberapa ruangan sekaligus. Hal ini dimungkinkan oleh adanya sistem basis data yang ditangani oleh sebuah PC yang dihubungkan

dengan sistem kunci ini melalui komunikasi serial.

8. LANGKAH-LANGKAH

- : - Persiapan
- Studi literatur
- Perencanaan
- Pembuatan alat
- Penulisan naskah

| LANGKAH KEGIATAN | B U L A N   K E |            |            |    |            |    |
|------------------|-----------------|------------|------------|----|------------|----|
|                  | I               | II         | III        | IV | V          | VI |
| PERSIAPAN        | ██████████      |            |            |    |            |    |
| STUDI LITERATUR  |                 | ██████████ |            |    |            |    |
| PERENCANAAN      |                 |            | ██████████ |    |            |    |
| PEMBUATAN ALAT   |                 |            | ██████████ |    |            |    |
| PENULISAN NASKAH |                 |            |            |    | ██████████ |    |

## BAB VI

### KESIMPULAN DAN SARAN

Sistem Kunci Elektronik ini dirancang untuk pengamanan pembukaan pintu dari satu arah, yaitu dari luar saja.

Dengan pemakaian sistem komunikasi serial pada sistem kunci elektronik ini, maka untuk komputer basis datanya dapat menggunakan segala macam komputer dengan syarat komputer tersebut harus dapat menyelenggarakan komunikasi serial dan dapat menangani program basis data.

Program basis data yang dibuat di dalam tugas akhir ini dapat diganti dengan program basis data yang lebih sesuai dengan keperluan pada pemakaiannya nanti.

Rancangan pintu disarankan menggunakan sistem elektrik. Tetapi hal ini tidaklah mutlak, rancangan mekanis mungkin saja digunakan dengan hasil yang memuaskan. Hal yang perlu diingat adalah bahwa rancangan pintu yang digunakan haruslah yang mempunyai kemampuan menutup sendiri. Peralatan ini tidak menyediakan kontrol untuk menutup pintu karena mekanisme ini harus sesuai dengan rancangan pintu yang digunakan.

Sistem kunci ini dapat dikembangkan untuk digunakan dalam sistem pengamanan lebih dari satu pintu dengan hanya menggunakan satu komputer basis data saja. Tentu saja untuk masing-masing pintu harus dipasang sebuah peralatan kunci elektronik.

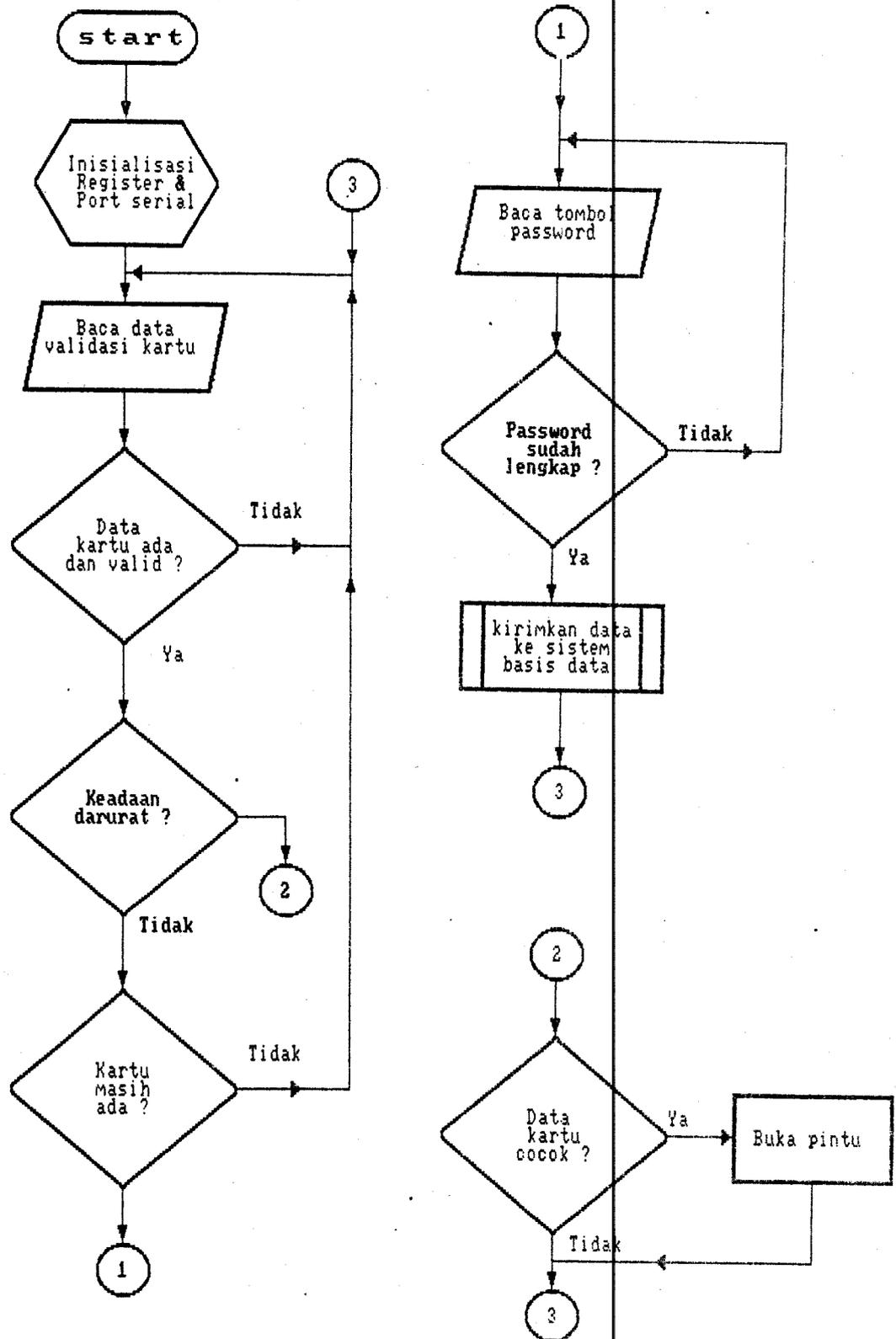
Program di dalam kontroller dan di komputer basis data belum dirancang untuk pemakaian lebih dari satu pintu. Untuk pemakaian sistem ini pada pengamanan lebih dari satu pintu, dibutuhkan tambahan program untuk identifikasi pintu. Disamping itu juga diperlukan juga peralatan pencabang komunikasi serial dari komputer basis data ke masing-masing kontroller (peralatan pencabang komunikasi serial sudah pernah dibuat dalam suatu tugas akhir di Jurusan Teknik Elektro FTI ITS).

## KEPUSTAKAAN

1. MCS-51<sup>R</sup> Family of single chip microcomputers user's manual, INTEL, January 1981
2. Turbo Pascal, Advance Programmer's Guide by Stephen K. O'Brien, Borland-Osborn-McGraw Hill, 1988
3. IBM-PC / PS2 Programmer's Guide, Peter Norton
4. Campbell. Joe. *The RS-232 Solution*. Barkley. California. 1988.
5. IBM. *Technical Reference for PC/XT System*. - . - .
6. Wyatt. Allen l.. *Using assembly Language*. - . - .
7. Hall. Douglas V.. *Microprocessors And Interfacing*, McGraw-Hill Book Co.. Inc. 1986.

# LAMPIRAN A

## FLOWCHART PROGRAM UNTUK PERANGKAT KERAS



## LAMPIRAN B

### PETUNJUK PEMAKAIAN PROGRAM PENANGANAN BASIS DATA

Program penanganan basis data yang dibuat disini ditujukan untuk dijalankan pada PC-IBM XT/AT. Program ini dibuat berdasarkan suatu asumsi yang berlaku pada sebuah model sistem pengamanan ruangan.

Ketika program mulai dijalankan, program akan mengambil data-data personil dari sebuah file, dimana demi keamanan sistem basis data file ini disimpan dalam keadaan tersandi. Setelah itu program akan masuk pada mode monitor, dimana program selalu memonitor kegiatan pada peralatan elektronik melalui komunikasi serial. Bila ada data yang diterima, maka program akan segera mencocokkan data yang diterima dengan data yang ada di dalam file basis data. Bila data tersebut ternyata cocok, maka program akan mengirimkan data ke peralatan elektronik agar membukakan pintu. Selain itu program juga mencatat kegiatan tersebut ke dalam sebuah file monitor yang juga tampak di layar.

Bila akan dilakukan pemeliharaan terhadap basis data, maka sebuah menu popup utama akan dapat diaktifkan dengan menekan tombol 'B'.

Pemeliharaan yang dapat dilakukan terhadap basis data meliputi penambahan, perubahan, penghapusan, pence-  
takan layout kartu personil, pembatalan / pemberian ijin

personil, dan pembuatan file basis data yang baru.

Setiap pemeliharaan yang dilakukan harus didahului dengan pemasukan password yang sesuai untuk tiap record yang bersangkutan. Password ini tentunya hanya diketahui oleh personil yang berkepentingan saja, yaitu personil itu sendiri dan personil yang menjabat sebagai supervisor dari sistem ini yang mempunyai password yang dapat digunakan untuk semua record.

Menu pemeliharaan basis data :

1. Tambah data personil baru. Tekan tombol 'T'.
2. Ubah data personil lama. Tekan tombol 'U'.
3. Hapus data satu personil. Tekan tombol 'H'.
4. Cetak layout kartu. Tekan tombol 'C'.
5. Pembatalan ijin Personil. Tekan tombol 'N'.
6. Pemberian ijin personil. Tekan tombol 'A'.
7. Pembuatan file basis data baru.
8. Perintah buka pintu.

(Menu 7 dan 8 hanya dapat dipilih oleh supervisor)

Sebagaimana sebuah menu popup biasanya, maka pilihan juga dapat dilakukan dengan menggerakkan bar pilihan ke menu yang diinginkan kemudian menekan tombol Enter. Gerakan bar pilihan ini akan diikuti dengan keterangan yang sesuai untuk pilihan yang bersangkutan. Keterangan ini terdapat di baris terbawah layar.

Bila sebuah menu pemeliharaan sudah dipilih, maka berikutnya akan ditanyakan field nama dari record yang dimaksud, kecuali pemasukan data personil baru atau pembuatan file basis data baru. Bila field nama dari record yang dimaksud dikosongi, maka akan dimunculkan menu popup untuk memilih record personil yang dimaksud.

Setelah didapat record yang dimaksud, maka akan ditanyakan password untuk record tersebut. Password yang dapat digunakan adalah password pemilik record ini sendiri ataupun password milik supervisor. Bila password dapat diterima oleh program, barulah dilakukan pemeliharaan sesuai dengan menu yang dipilih sebelumnya. Bila password salah, maka pemeliharaan tidak dapat dilakukan.

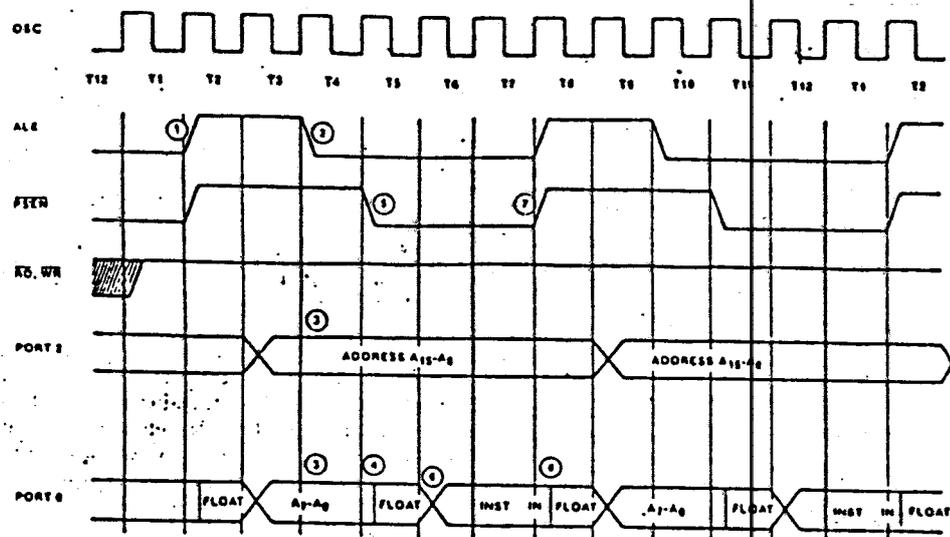
Untuk maksud perubahan status perijinan personil ataupun yang lain, pada saat pemeliharaan basis data dilakukan program tidak lagi berada pada mode monitor. Dengan demikian kegiatan peralatan elektronik tidak diikuti lagi oleh program, dengan pertimbangan pada saat itu pintu tidak boleh dibuka untuk sementara.

Program dapat dikembalikan lagi ke mode monitor dengan menonaktifkan menu popup pemeliharaan basis data. Untuk itu dapat dilakukan dengan menekan tombol ESC.

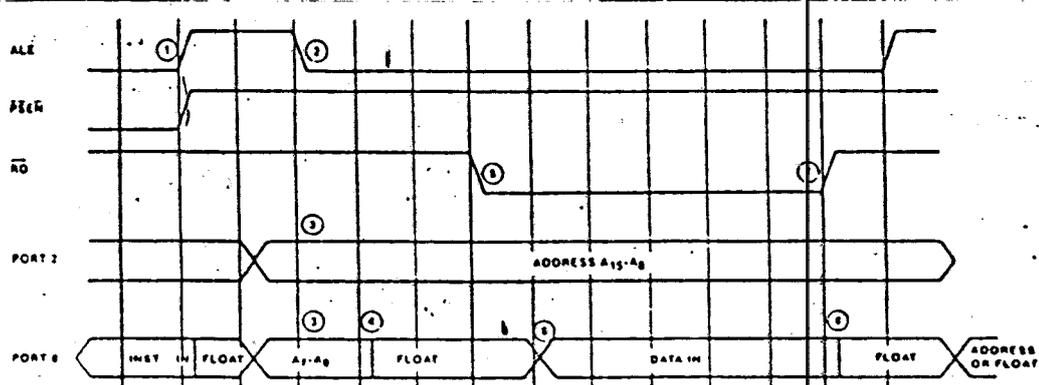
Program dapat diakhiri untuk kembali ke sistem

operasi dengan menekan tombol 'S' dari mode monitor.

Input dari Three State Buffer dihubungkan dengan peralatan pembaca kode kartu dan pembaca tombol password. Lihat gambar 3.4. Dengan cara ini data dari peralatan pendukung diperlakukan sebagai data dari RAM luar.



Gambar 3.1 Program memory read cycle timing



Gambar 3.2 Data memory read cycle timing

Port 1 dipakai untuk mengirimkan data ke peralatan-peralatan pendukung, yaitu pembaca kode kartu, pembaca tombol password, dan pengendali peralatan mekanik kunci pintu, serta pemberi tanda penerimaan pembacaan tombol.

### III.2 Peralatan pendukung

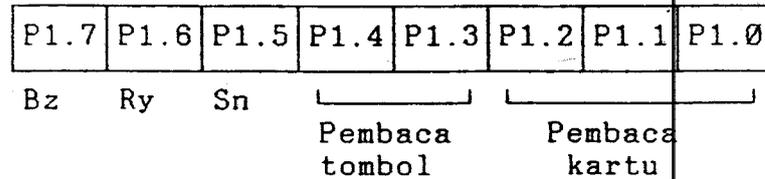
Peralatan pendukung ditangani oleh port 1 dan port 0. Port 1 memberikan data-data kendali ke peralatan-peralatan pembaca kode kartu (P1.0 - P1.2), pembaca tombol password (P1.3, P1.4), pengendali peralatan mekanik kunci pintu (P1.5, P1.6), dan pemberi tanda penerimaan pembacaan tombol (P1.7). Data ini merupakan data scan untuk operasi pembacaan kode kartu dan pembacaan tombol password.

#### III.3.2.1 Pembaca kartu

Pada pembacaan kartu, port 1 mengirimkan data scan untuk mengaktifkan sensor optik yang berupa fototransistor.







Gambar 3.5  
Pemakaian port 1

Untuk membaca kode tiga byte yang ada pada setiap kartu, sensor-sensor optik ditempatkan dalam matrix 3 x 8. Satu baris untuk validasi kartu, dan dua yang lain untuk pembaca kode kartu. Sensor-sensor tersebut mendeteksi sinar dari tiga baris pemancar. Pengaktifan dilakukan dengan menghubungkan sebaris sensor dengan jalur catu daya melalui sebuah transistor. Sensor yang menerima sinar dari pemancar akan memberikan logika 1 kepada masukan Three State Buffer.

### III.3.2.2 Pembaca tombol password

Pembacaan tombol password memakai prinsip yang sama dengan pembaca kartu. Tombol tekan yang disusun dalam Matrix 8 x 2 digunakan untuk menghubungkan masukan Three State Buffer dengan jalur catu positif. Penekanan tombol akan memberikan logika 1 kepada masukan Three State Buffer. Dengan cara ini penekanan tombol hanya berpengaruh terhadap input dari port 1 pada saat yang diinginkan.

### III.3.2.3 Pengendali sistem mekanik

Peralatan berikutnya adalah pengendali mekanik kunci pintu. Pengendali ini berupa relay yang dapat dihubungkan dengan sistem mekanik yang digerakkan listrik, sedangkan mekanik selanjutnya dapat dipilih sesuai kebutuhan. Mekanik yang disarankan adalah yang dapat secara otomatis menutup pintu kembali setelah membukanya. P1.5 dihubungkan dengan saklar ditempatkan di sistem mekanik pintu. Saklar ini digunakan sebagai masukan untuk mendeteksi pintu sedang tertutup atau terbuka.

### III.3.2.4 Pemberi tanda penerimaan penekanan tombol password

Pemberi tanda penekanan tombol password adalah sebuah buzzer peizo-elektrik yang diaktifkan oleh P1.7. Buzzer ini memberikan suatu bunyi pendek ketika penekanan suatu tombol password telah diterima sebagai data yang sah.

### III.3.3 Catu daya

Catu daya dirancang untuk mendukung sistem ini agar tetap dapat bekerja pada waktu listrik dari sumber jala-

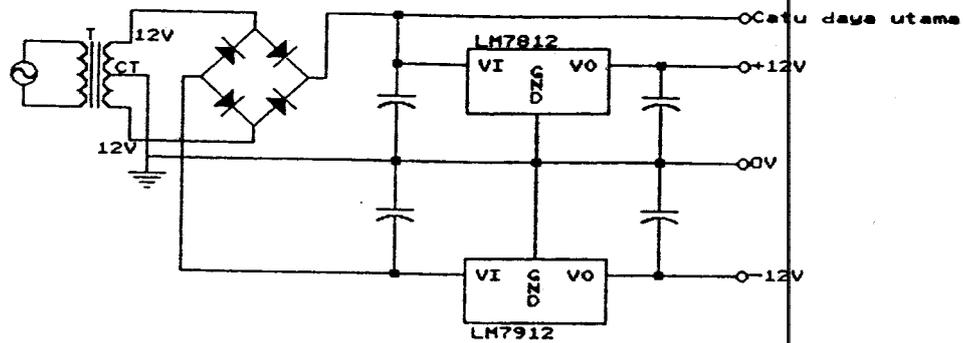
jala tidak dapat memberikan daya.

Pada keadaan normal sistem dioperasikan dengan daya yang didapat dari sebuah adaptor catu daya yang menghasilkan tegangan  $\pm 12V$  dan  $0V$ . Catu ini distabilkan oleh IC 7812 dan IC 7912 yang dipakai untuk mencatu daya chip gerbang komunikasi serial. Catu daya juga dihubungkan dengan pengisi baterai +9V dan IC pengatur tegangan dengan output +5 V untuk mencatu daya IC-IC TTL di sistem utama. Baterai +9V digunakan sebagai cadangan pada saat catu daya utama mengalami kegagalan. Lihat gambar 3.6, dan 3.7.

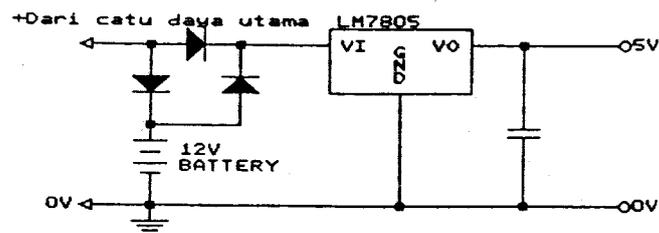
Pada saat catu daya utama mengalami kegagalan, baterai akan mengambil alih tugasnya. Catu daya dari baterai hanya akan memberikan daya melalui IC regulator kepada sistem utama. Sedangkan catu daya untuk gerbang komunikasi serial tidak diberikan, karena hanya didapat dari catu daya utama. Hal ini menimbang bahwa kegagalan catu daya utama terutama disebabkan oleh adanya gangguan terhadap sumber daya jala-jala. Dengan demikian komputer basis data pun sangat mungkin juga mengalami gangguan.

Untuk menjaga agar pintu tetap dapat dibuka, maka sistem utama harus tetap bekerja tanpa basis data. Dalam keadaan ini pintu hanya bisa dibuka dengan kartu dan

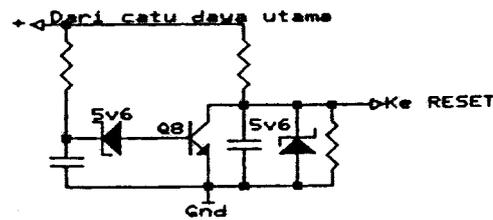
password khusus, sehingga basis data tidak diperlukan.



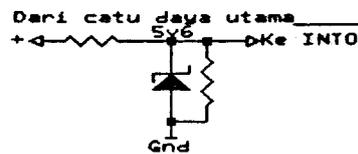
Gambar 3.6  
Rangkaian catu daya utama



Gambar 3.7  
Rangkaian catu daya darurat dan  
regulator tegangan sistem utama



Gambar 3.8  
Rangkaian reset otomatis



Gambar 3.9  
Rangkaian detektor kegagalan catu daya

Pin external interrupt  $\overline{INT0}$  diprogram pada low level activated untuk mendeteksi terjadinya kegagalan terhadap catu daya utama. Rutin pelayanan dari interrupt ini akan memberikan informasi kepada sistem untuk bekerja secara darurat. Bila kemudian catu daya utama bisa berfungsi kembali, maka sistem direset kembali secara otomatis dan bekerja seperti semula. Lihat gambar 3.8 dan 3.9.

## BAB IV

### RANCANGAN PERANGKAT LUNAK

#### IV.1 Pendahuluan

Perangkat lunak kontroller disusun dalam bahasa program assembler MCS-51. Di dalamnya terdapat 51 buah instruksi dasar yang dibagi lagi menjadi lima kelompok fungsi. Ditambah lagi dengan beberapa mode pengalamatan yaitu jenis data boolean (1 bit), nibble (4 bit), byte (8 bit) dan alamat (16 bit) menghasilkan 111 buah instruksi.

Setiap instruksi program mengandung sebuah kata-kata operasi dan sampai empat buah *operand* (tergantung pada instruksinya). Instruksi yang memerlukan dua data operand selalu menjadikan yang pertama sebagai penerima hasil operasi, kecuali bila kedua operand dalam mode *direct*. Dalam kasus ini operand kedualah yang menjadi penerima hasil operasi.

Memori diorganisasikan menjadi tiga ruang alamat dan sebuah program counter. Ruang memori dapat dilihat pada gambar 4.1. Ruang-ruang tersebut adalah sbb.:

- 16 bit program counter
- 64K-byte total ruang alamat memori program